Model Curriculum for UG Degree Course in

Computer Science and Engineering

(Engineering & Technology)

2022

ALL INDIA COUNCIL FOR TECHNICAL EDUCATION

Nelson Mandela Marg, Vasant Kunj, New Delhi 110070

www.aicte-india.org

Revised Model Curriculum for
UG Degree Course
in
Computer Science and Engineering
(Engineering & Technology)

ALL INDIA COUNCIL FOR TECHNICAL EDUCATION

NELSON MANDELA MARG, Vasant Kunj, New Delhi – 110070

www.aicte-india.org

Committee for Model Curriculum

S.No	Name	Designation & Organization
1	Dr. Pankaj Jalote	Distinguished Professor and founding Director, IIIT-Delhi. (Chairman)
2	Dr. Dheeraj Sanghi	Professor, Computer Science and Engineering Dept., PEC, Chandigarh
3	Dr. Manoj Singh Gaur	Director, Computer Science and Engineering Dept., IIT Jammu
4	Dr. Nutan Limaye	Professor, Computer Science and Engineering Dept., IIT Bombay
5	Dr. Ramkumar Ramamoorthy	Former Chairman and MD, Cognizant India
6	Mr. Amit Aggarwal	NASSCOM

**This committee was responsible for the design of the CSE curriculum and its courses only. Other committees have designed the common program and non-CSE courses. Complete list of experts who contributed to the design of the CSE curriculum is given at the end.*

MESSAGE

The quality of technical education depends on many factors but largely on- outcome based socially and industrially relevant curriculum, good quality motivated faculty, teaching learning process, effective industry internship and evaluation of students based on desired outcomes. Therefore, it was imperative that a Model Curriculum be prepared by best experts from academia and industry, keeping in view the latest industry trends and market requirements and be made available to all universities / board of technical education and engineering institutions in the country. AICTE constituted team of experts to prepare the model curriculum of UG Degree Course in Computer Science and Engineering. Similar exercise is done for other UG, Diploma and PG level in engineering, MBA, PGDM, Architecture, etc.

It comprises of basic science and engineering courses, having focus on fundamentals, significant discipline level courses and ample electives both from the disciplines and cross disciplines including emerging areas all within a cumulative structure of 163 credits. Summer Internships have been embedded to make the student understand the industry requirements and have hands on experience. Virtual Labs has been introduced for few experiments. Also, most courses have been mapped to its equivalent SWAYAM/NPTEL Course to offer an alternative for learning that course online from SWAYAM. These features will allow students to develop a problem-solving approach to face the challenges in the future and develop outcome based learning approach.

As a major initiative by AICTE, a three-week mandatory induction program for students has also been designed and has to be given at the beginning of the course. The idea behind this is to make the students feel comfortable in their new environment, open them up, set a healthy daily routine, develop awareness, sensitivity and understanding of the self, people around them, society at large, and nature.

AICTE places on record, special thanks to Dr. Pankaj Jalote, Dr. Dheeraj Sanghi, Dr. Manoj Singh Gaur, Dr. Nutan Limaye, Dr. Ramkumar Ramamoorthy and other committee members. We are sure that this Model Curriculum will help to enhance not just the employability skills but will also enable youngsters to become job creators.

We strongly urge the institutions / universities / boards of technical education in India to adopt this Model Curriculum at the earliest. This is a suggestive curriculum and the concerned university / institution / board should build on and exercise flexibility in readjustment of courses within the overall 163 credits.

(Prof. M. Jagadesh Kumar)
Chairman
All India Council for Technical Education

PREFACE

Taking cognizance of growing concern about quality of technical education in India, AICTE in its 49th council meeting held on 14.03.2017 approved a package of measures for improving quality of technical education - Revision of Curriculum, Mandatory Internship, and Student Induction Program were amongst the few.

AICTE constituted committee of academia industry experts to prepare model curriculum of UG Course in Computer Science and Engineering. During the development of curriculum, the employability and employment opportunities for graduates, future ready workforce who will be skilled enough to handle the rapid growth in the field of Computer Science and Engineering were kept in mind.

AICTE has introduced mandatory internship in the new curriculum which will equip the students with practical understanding and training about industry practices in a suitable industry or organization. In the course of development of model curriculum, the committee took feedback of industry experts on the draft curriculum and accordingly modified the draft before finalization. This exercise has ensured that essential emphasis on industry requirements and market trends, employability and problem solving approach is given.

After due deliberations, the scheme and syllabus have been formulated. Salient features of this model curriculum are enumerated as under:

- Reduced number of credits.
- Introduction of Student Induction Program.
- Well defined learning objectives & outcomes for each course.
- Inclusion of courses on socially relevant topics.
- Built-in flexibility to the students in terms of professional elective and open elective courses.
- Mandatory internship to equip the students with practical knowledge and provide them exposure to real time industrial environments.
- Virtual Labs.
- Mapping of Courses to its equivalent NPTEL/SWAYAM Course.
- Course on 'Entrepreneurship and Startups' to encourage entrepreneurial mindset.
- Introduction of Design Thinking and Universal Human Value course.

Thanks for the time and efforts of the members of the working group Chaired by Prof. Pankaj Jalote and which included -Dr. Dheeraj Sanghi, Dr. Manoj Singh Gaur, Dr. Nutan Limaye, Dr. Ramkumar Ramamoorthy and other committee members.

Special thanks to Prof. Anil D. Sahasrabudhe, Chairman; Prof. M.P. Poonia, Vice-Chairman; and Prof. Rajive Kumar, Member Secretary, AICTE who all have been instrumental and encouraging throughout the process of development of this model curriculum, and the dedicated efforts of Dr. Naveen Arora, Assistant Director (P&AP);

Dr. Anil Sharma, Assistant Director (P&AP), Mr. Rakesh Kumar Pandit, Young Professional (P&AP); Ms. Nishtha Sehgal, IT Consultant and other office staff of AICTE.

Dr. Ramesh Unnikrishnan
Advisor – II (P&AP)

Table of Contents

GENERAL COURSE STRUCTURE & CREDIT DISTRIBUTION

GENERAL COURSE STRUCTURE & THEME

A. Definition of Credit:

1 Hr. Lecture (L) per week	1 Credit
1 Hr. Tutorial (T) per week	1 Credit
1 Hr. Practical (P) per week	0.5 Credit
2 Hours Practical (P) per week	1 Credit

B. Range of Credits: In the light of the fact that a typical Model Four-year Under Graduate degree program in Engineering has about 163 credits, the total number of credits proposed for the four-year B. Tech/B.E. in Computer Science and Engineering (Engineering & Technology) is kept as 163.

C. Structure of UG Program in CSE: The structure of UG program in Computer Science and Engineering shall have essentially the following categories of courses with the breakup of credits as given:

S. No.	Category	Credit Breakup for CSE students
1	Humanities and Social Sciences including Management courses	16
2	Basic Science courses	23
3	Engineering Science courses including workshop, drawing, basics of electrical/mechanical/computer etc.	29
4	Professional core courses	59
5	Professional Elective courses relevant to chosen specialization/branch	12
6	Open subjects – Electives from other technical and /or emerging subjects	9
7	Project work, seminar and internship in industry or elsewhere	15
8	Mandatory Courses [Environmental Sciences, Induction Program, Indian Constitution, Essence of Indian Knowledge Tradition]	(non-credit)
	Total	**163***

**Minor variation is allowed as per need of the respective disciplines.*

D. Course code and definition:

Course code	Definitions
L	Lecture
T	Tutorial
P	Practical
C	Credits
BSC	Basic Science Courses
ESC	Engineering Science Courses
HSMC	Humanities and Social Sciences including Management courses

PCC-CS	Professional core courses
PEC -CS	Professional Elective courses
OEC-CS	Open Elective courses
LC	Laboratory course
MC	Mandatory courses

- **Course level coding scheme:** Three-digit number (odd numbers are for the odd semester courses and even numbers are for even semester courses) used as suffix with the Course Code for identifying the level of the course. Digit at hundred's place signifies the year in which course is offered. e.g.

  101, 102 … etc. for first year.

  201, 202 …. Etc. for second year.

  301, 302 … for third year.

- **Category-wise Courses**

HUMANITIES & SOCIAL SCIENCES COURSES [HS]

(i) Number of Humanities & Social Science Courses: 6
(ii) Credits: 16

Sl. No	Code No.	Course Title	Semester	Hours per week			Total Credits
				Lecture	Tutorial	Practical	
1	HSMC-201	English	I	2	0	2	3
2	HSMC-102	Design Thinking	I	0	0	2	1
3	HSMC (H-102)	Universal Human Values-II: Understanding Harmony And Ethical Human Conduct	II	2	1	0	3
4	HSMC-301	Humanities – 1	III	3	0	0	3
5	HSMC-401	Management-I (Organizational Behaviour)/ Finance & Accounting	IV	3	0	0	3
6	HSMC-501	Humanities – II	V	3	0	0	3
Total Credits							**16**

****

BASIC SCIENCE COURSE [BSC]

Sl. No	Code No.	Course Title	Semester	Hours per week			Total Credits
				Lecture	Tutorial	Practical	
1	BSC-101	Physics-I (Semi-conductor Physics)	I	3	1	2	5
2	BSC-201	Mathematics-II (Probability and Statistics)	II	3	1	0	4
3	BSC-102	Mathematics-I (Calculus and Linear Algebra)	I	3	1	0	4
4	BSC-202	Chemistry-I	II	3	1	2	5
5	BSC-701	Biology	VII	2	1	0	3
6	BSC-301	Mathematics-III (Differential Calculus)	III	2	0	0	2
Total Credits							**23**

****

ENGINEERING SCIENCE COURSE [ESC]

Sl. No	Code No.	Course Title	Semester	Hours per week			Total Credits
				Lecture	Tutorial	Practical	
1	ESC-101	Basic Electrical Engineering	I	3	1	2	5
2	ESC-102	Engineering Graphics & Design	I	1	0	4	3
3	ESC-201	Programming for Problem Solving	II	3	0	4	5
4	ESC-202	Workshop/Manufacturing Practices	II	1	0	4	3
5	ESC-301	Analog Electronic Circuits	III	3	0	4	5
6	ESC-302	Digital Electronics	III	3	0	4	5
7	ESC-501	Signals and Systems	V	3	0	0	3
Total Credits							**29**

****

PROFESSIONAL CORE COURSES [PCC]

S No	Code No.	Course Title	Semester	Hours per week			Total Credits
				Lecture	Tutorial	Practical	
1	PCC CS-301	Data Structure and Algorithms	III	3	0	4	5
2	PCC CS-302	IT Workshop – (Sci Lab/MATLAB)	III	1	0	4	3
3	PCC CS-401	Discrete Mathematics	IV	3	1	0	4
4	PCC CS-402	Computer Organization and Architecture	IV	3	0	4	5
5	PCC CS-403	Operating Systems	V	3	0	4	5
6	PCC CS-404	Design and Analysis of Algorithms	IV	3	0	4	5

7	PCC CS-405	Advanced Programming	IV	3	1	0	4
8	PCC CS-505	Introduction to Database Systems	V	3	0	4	5
9	PCC CS-603	Machine Learning	V	3	1	0	4
10	PCC CS-504	Theory of Computation	VI	3	1	0	4
11	PEC CS-601	Introductory Cyber Security	VI	3	0	4	5
12	PCC CS-601	Computer Networks	VI	3	0	4	5
13	PCC CS-602	Complier Design	VII	3	0	4	5
Total Credits							**59**

****

PROFESSIONAL ELECTIVE [PEC]

S. No	Code No.	Course Title	Semester	Hours per week			Total Credits
				Lecture	Tutorial	Practical	
1	PEC	Elective – I	VI	3	0	0	3
2	PEC	Elective-II	VI	3	0	0	3
3	PEC	Elective-III	VII	3	0	0	3
4	PEC	Elective-IV	VIII	3	0	0	3
Total Credits							**12**

****

OPEN ELECTIVE COURSES [OEC]

S.No	Code No.	Course Title	Semester	Hours per week			Total Credits
				Lecture	Tutorial	Practical	
1	OEC	Open Elective – I	VII	3	0	0	3
2	OEC	Open-Elective-II	VIII	3	0	0	3
3	OEC	Open-Elective-III	VIII	3	0	0	3
Total Credits							**9**

****

PROJECT WORK, SEMINAR AND INTERNSHIP IN INDUSTRY OR ELSEWHERE

Sl. No	Course Code	Course Title	Semester	Hours per week			Total Credits
				Lecture	Tutorial	Practical	
1	PROJ CS-601	Project-I	VI	0	0	6	3
2	PROJ CS-701	Project-II	VII	0	0	12	6
3	PROJ CS-801	Project-III	VIII	0	0	12	6
Total Credits							**15**

******

INDUCTION PROGRAM

The Essence and Details of Induction program can also be understood from the 'Detailed Guide on Student Induction program', as available on AICTE Portal, (Link:https://www.aicteindia.org/sites/default/files/Detailed%20Guide%20on%20Student%20Induction%20program.pdf). For more, Refer **Appendix II.**

Induction program (mandatory)	**Three-week duration**
Induction program for students to be offered right at the start of the first year.	• Physical activity • Creative Arts • Universal Human Values • Literary • Proficiency Modules • Lectures by Eminent People • Visits to local Areas • Familiarization to Dept./Branch & Innovations

E. Mandatory Visits/ Workshop/Expert Lectures:

a. It is mandatory to arrange one industrial visit every semester for the students of each branch.

b. It is mandatory to conduct a One-week workshop during the winter break after fifth semester on professional/ industry/ entrepreneurial orientation.

c. It is mandatory to organize at least one expert lecture per semester for each branch by inviting resource persons from domain specific industry.

F. Evaluation Scheme (Suggestive only):

a. **For Theory Courses:**
(The weightage of Internal assessment is 40% and for End Semester Exam is 60%)
The student has to obtain at least 40% marks individually both in internal assessment and end semester exams to pass.

b. **For Practical Courses:**
(The weightage of Internal assessment is 60% and for End Semester Exam is 40%)
The student has to obtain at least 40% marks individually both in internal assessment and end semester exams to pass.

c. For Summer Internship / Projects / Seminar etc.
Evaluation is based on work done, quality of report, performance in viva-voce, presentation etc.

Note: The internal assessment is based on the student's performance in mid semester tests (two best out of three), quizzes, assignments, class performance, attendance, viva-voce in practical, lab record etc.

G. Mapping of Marks to Grades

Each course (Theory/Practical) is to be assigned 100 marks, irrespective of the number of credits, and the mapping of marks to grades may be done as per the following table:

Range of Marks	**Assigned Grade**
91-100	AA/A^{+}
81-90	AB/A
71-80	BB/B^{+}
61-70	BC/B
51-60	CC/C^{+}
46-50	CD/C
40-45	DD/D
< 40	FF/F (Fail due to less marks)
-	F^{R} (Fail due to shortage of attendance and therefore, to repeat the course)

****

Framework for CSE Curriculum Design

Context and Background

Computer Science has changed considerably in the last few years with areas like Machine Learning and Cloud computing becoming much more important. At the same time, the technologies and underlying computing systems have also evolved, considerably easing executing some of the tasks that earlier took much more training and experience. These changes require that teaching of computer science ought to suitably adapt – to reflect the changed nature of the discipline, as well as to update courses with the more recent technology platforms. And this ought to be done while providing some flexibility to the HEIs to address their specific constraints and focus. With these in mind, the committee for model curriculum design for CSE established some of the guiding principles for the exercise:

- The focus of curriculum design is the 4 year BE/B Tech program in CSE and the target audience of the curriculum is the vast majority of universities/institutions, rather than the premier institutes (who design their own curriculum and have the expertise for it.)
- The existing CSE curriculum of AICTE will be the starting point. Suitable enhancements / modifications will be made.
- **Flexibility**. Even within the large non-top tier education system, which is the primary target, some have better infrastructure and capability, so it is desirable to provide a limited degree of flexibility to the HEIs on the curriculum.

  To provide this flexibility, for each course, the learning outcomes will be grouped in two – essential, and desired/advanced. The core courses will also be identified as essential and desired/advanced.

  This simple framework of grouping outcomes and courses as essential and desired provides limited flexibility to institutions to design their curriculum depending on their capabilities, resources, goals, etc. while still providing guidance for a sound curriculum. The flexibility can be leveraged by institutions to provide different pathways to students, and multiple exits.
- **Early Exits**. The new education policy (NEP) aims to provide multiple exit points to students. To support more than one exits for students, it is important to develop employability skills early, and not take the approach of first focusing on foundations and then develop practical skills. Also, we believe that all but a few students will go for full degree, hence while providing flexibility for early exit, the outcomes of the 4-year degree should not be compromised. We propose:
  - Eliminate the separation of theory and labs, instead break courses topic or theme-wise. This will allow both basic theory and basic skills to be taught together and build skills early.
  - Introduce discipline courses early, so disciplinary skills can be developed early. For this have one sem common program, and allow some discipline courses in 2nd sem – this will also help students in getting internships / part-time jobs in summer. This will require branch change to be done after one semester. (For lateral entry, some make-up courses may be needed.)

  With this, and the flexibility provided in the curriculum, an HEI can design suitable exits. We explicitly provide a recommendation for possible exits for a B Tech CSE student.
- **Provide guidance for labs/assignments.** A small analysis the committee did on usage of AICTE curriculum indicated that while the courses and topics specified in the curriculum are widely used, HEIs differ considerably in types of assignments/labs/projects they give. Also, this is where there is a perceived deficiency in education – while theory is covered suitable assignments/labs are not give

for students to develop skills. To help in this, it was decided that for each course design, experts will also recommend nature of labs and assignments for each module in the course.

- **Pedagogy suggestions**. With the emergence of a host of online resources that can be used to support teaching, there may be different ways to teach a topic. To help teachers in pedagogy, for each module in a course, pedagogy suggestions have also been provided.
- **Multiple pathways**. For supporting multiple pathways within the academic program, we propose to provide for micro specializations through thematic course streams. These can be further enhanced by HEIs with programs like honors for advanced students with more credits or advanced learning outcomes, etc.

Graduate Attributes

Curriculum of a program is finally a network of credit units – courses (core, disciplinary core, disciplinary elective, open), internships, practice, projects, etc. which help achieve program goals. Program goals can be stated as attributes the students should possess on graduation, i.e. statements about the learning, values, capabilities etc. of graduates. These are called Graduate Attributes (GAs). A program typically has:

- **General GAs:** which are often common across many similar programs (e.g. B Techs) and focus on generalized skills and capabilities in the graduate.
- **Discipline GAs**: are discipline specific attributes, which focus on understanding of different concepts and systems related to the discipline, and on competencies and skills in that discipline.

Together the GAs define the goals of the program. The aim of curriculum design is to evolve a curriculum that can develop in students the stated graduate attributes. While specifying the GAs and designing a curriculum for it, we must keep a basic constraint in mind: a full B Tech program has 8 semesters, each with about 5 full courses. GAs should specify only what can be taught and absorbed in this time box (i.e. we cannot push more simply by adding more.)

Desired Graduate Attributes for the CSE program are given below. The CSE curriculum design will focus more on delivering the discipline GAs, while strengthening the general GAs, where possible. Feedback on these Graduate Attributed was taken from many representatives from industry, as well as from HEIs. GAs should be read by adding this at the start of each: At graduation time, a student should have:

General Graduate Attributes	Discipline Graduate Attributes
G1 Ability to identify a problem, analyze using design thinking techniques, and evolve innovative approaches for solving it.	CS1 Proficiency in writing in at least two dissimilar programming languages programs of modest complexity which are: readable, tested for correctness, efficient, and secure.
G2 Ability to apply mathematical concepts and techniques in problem solving.	CS2 Ability to design and apply appropriate algorithms and data structures for evolving efficient computing based solutions for new problems.

G3 Ability to function effectively in multi-cultural teams to accomplish a common goal.	CS3 Understanding of computing systems at computer architecture, operating systems, and distributed- computing levels, and how they affect the performance of software applications.
G4 Ability to communicate effectively with a wide range of audience.	CS4 Understanding of theoretical foundations, fundamental principles, and limits of computing.
G5 Ability to self-learn and engage in life-long learning and upgrade technical skills	CS5 Ability to analyse large volumes of data employing a variety of techniques for learning, better prediction, decision making, etc.
G6 An understanding of professional and ethical responsibility	**ADVANCED/OPTIONAL GAs** CS6 Ability to design, implement, and evaluate computer based system or application to meet the desired needs using modern tools and methodologies
G7 Ability to undertake small research tasks and projects.	CS7 Ability to develop full stack applications using one commonly used tech-stack and modern tool.
G8 An entrepreneurial mind set for opportunities using technology and innovations.	CS8 Understanding of and ability to use advanced techniques and tools in a few different domain areas (e.g. parallel processing, image processing, IR, …)
G9 An understanding of impact of solutions on economic, societal, and environment context.	CS9 Exposure to emerging technologies such as Cloud Computing, IoT, etc.
G10 Strong emotional intelligence, human and cultural values	

CSE Core Courses

For the B Tech CSE program, students need to do as part of their common 1st year curriculum a course on programming ("Introduction to Programming") and two Math courses.

The list of core courses recommended for CSE are given below. As explained above, the core has been split into two – essential and additional. The essential core is what all programs must have. The additional core list those courses which HEIs can choose to make core or an elective, depending on their education approach and resources.

Suggested Year	**Course Name**	**GAs it Contributes to (GAs it supports)**
	Professional Core Courses (Essential)	
2nd year	Data structures and Algorithms (DS)	CS1, CS2, CS4
2nd year	Discrete Mathematics (DM)	CS4 (CS2, CS4, CS6)
2nd year	Computer Organization and Architecture (COA)	CS3 (CS4, CS6)
2nd year	Advanced Programming (AP)	CS1, CS3, CS7, CS8 (CS2)
3rd year	Operating Systems (OS)	CS3 (CS6, CS8)
3rd year	Design and Analysis of Algorithms (Algo)	CS1, CS2, CS4
3rd year	Database Systems (DB)	CS3, CS5, CS6 (CS1, CS7, CS9)
3rd/4th year	Computer Networks (NW)	CS3 (CS4, CS6, CS8)
3rd year	Machine Learning (New) (ML)	CS5,CS6,CS8
3rd year	Introductory Cyber Security (New)	CS1, CS3, CS6
3rd/4th year	Compiler Construction/Design	CS2, CS3, CS4
3rd/4th year	Theory of Computation	CS4 (CS2)

Detailed syllabus of each of these courses follows – a separate subsection for each course. A common template has been used to specify the course design. Each course design lists essential learning outcomes for the course, and desired/advanced learning outcomes. As the terms suggest, all HEIs should ensure that the essential learning outcomes are achieved. The desired/advanced learning outcomes are those which those HEIs which have the capabilities and resources to deliver may include in their courses.

Coverage of CS Graduate Attributes by the Core Courses

This table summarizes which CS GAs different courses contribute to. For each GA, list of main courses is given – these are the courses which directly contribute to the GA. Also mentioned are the courses which support the GA (though perhaps less directly):

CS1 Proficiency in writing in at least two dissimilar programming languages programs of modest complexity which are: readable, tested for correctness, efficient, and secure	Main Courses: DS, AP, Algo, Security Supporting Courses: DB
CS2 Ability to design and apply appropriate algorithms and data structures for evolving efficient computing based solutions for new problems	Main: DS, Algo Supporting: DM, AP, ToC
CS3 Understanding of computing systems at computer architecture, operating systems, and distributed-computing levels, and how they affect the performance of software application	Main: CO/CA, AP, OS, DB, NW, Security Supporting: Compilers
CS4 Understanding of theoretical foundations, fundamental principles, and limits of computing	Main: DS, DM, Algo, ToC Supporting: NW, CO/CA
CS5 Ability to analyse large volumes of data employing a variety of techniques for learning, better prediction, decision making,	Main: DB Supporting: DM
ADVANCED/OPTIONAL CS6 Ability to design, implement, and evaluate computer based system or application to meet the desired needs using modern tools and methodologies	Main: AP, DB, Security Supporting: DM, CO/CA
CS7 Ability to develop full stack applications using one commonly used tech-stack and modern tools	Main: AP Supporting: OS, NW, Security
CS8 Understanding of and ability to use advanced techniques and tools in a few domain areas (e.g. parallel processing, image processing, IR, …)	Main: Supporting: DB
CS9 Exposure to emerging technologies such as Cloud Computing, IoT, etc.	Main: Supporting: DB, Security, NW

Micro Specializations (and Professional Electives)

Besides the core courses, programs normally have professional elective courses, which HEIs decide. It is possible to use the electives to provide a limited specialization in some sub-area of CSE to a B Tech student. We call these as micro-specializations. These also allow multiple pathways to students, as different students can graduate with different specializations (or not). A micro specialization is a set of a few full or half courses, which build upon the core CSE program. The report gives possible structure of a few micro specializations in:

- Software Engineering
- Machine Learning
- Distributed and Cloud Systems
- Human Computer Interaction (HCI)
- Advanced Mobile Communications / 5G Micro Specialization

More specializations definition may be added (e.g. in Security, High Performance Computing, Algorithms). Micro specializations are optional for HEIs – i.e. they can decide to offer them or not, and if they do, which ones they want to. If they decide not to have these, the list of courses mentioned in the micro specializations can be used as a list of suggested professional electives.

Recommendations for Online Credits

A sub-committee was formed to recommend how online credits may be given by HEIs. The recommendations are given as part of the report as an appendix. The recommendations were accepted by the full steering committee.

Recommendations for Multiple Exits

A sub-committee was formed to recommend possible exits for a B Tech CSE student – in line with one of the goals of the NEP. The recommendations are given as part of the report as an appendix. The recommendations were accepted by the full steering committee.

****

Semester wise Structure and Curriculum for

UG Course in

Computer Science and Engineering (Engineering & Technology)

Semester I						
3-Week Orientation Programme						
S.No	**Course Code**	**Course Title**	**L**	**T**	**P**	**Credits**
1.	BSC-101	Physics-I	3	1	2	5
2.	BSC-102	Mathematics-I	3	1	0	4
3.	ESC-101	Basic Electrical Engineering	3	1	2	5
4.	ESC-102	Engineering Graphics & Design	1	0	4	3
5.	HSMC-102	Design Thinking	0	0	2	1
6.	AU-101^	IDEA Lab Workshop	2	0	4	0
Total						**18**
Note: ^ represents "Audit Course".						

Semester II						
S.No	**Course Code**	**Course Title**	**L**	**T**	**P**	**Credits**
1.	BSC-202	Chemistry-I	3	1	2	5
2.	BSC-201	Mathematics-II	3	1	0	4
3.	ESC-201	Programming for Problem Solving	3	0	4	5
4.	HSMC-201	English	2	0	2	3
5.	ESC-202	Workshop/Manufacturing Practices	1	0	4	3
6.	AU-102^	Sports and Yoga or NSS/NCC	2	0	0	0
7.	HSMC(H-102)	Universal Human Values-II: Understanding Harmony And Ethical Human Conduct	2	1	0	3
Total						**23**
Note: ^ represents "Audit Course".						

Semester III						
S.No	**Course Code**	**Course Title**	**L**	**T**	**P**	**Credits**
1.	ESC-301	Analog Electronic Circuits	3	0	4	5
2.	PCC CS-301	Data structure and Algorithms	3	0	4	5
3.	ESC-302	Digital Electronics	3	0	4	5
4.	PCC CS-302	IT Workshop (Sci Lab/MATLAB)	1	0	4	3
5.	BSC-301	Mathematics-III (Differential Calculus)	2	0	0	2
6.	HSMC-301	Humanities-I	3	0	0	3
Total						**23**

Semester IV						
S.No	**Course Code**	**Course Title**	**L**	**T**	**P**	**Credits**
1.	PCC CS-401	Discrete Mathematics	3	1	0	4
2.	PCC CS-402	Computer Organization & Architecture	3	0	4	5
3.	PCC CS-404	Design & Analysis of Algorithms	3	0	4	5
4.	PCC CS-405	Advanced Programming	3	1	0	4
5.	HSMC-401	Management 1 (Organizational Behaviour/ Finance & Accounting)	3	0	0	3
6.	MC	Environmental Sciences	-	-	-	0
Total						**21**

Semester V						
S.No	**Course Code**	**Course Title**	**L**	**T**	**P**	**Credits**
1.	ESC-501	Signals & Systems	3	0	0	3
2.	PCC CS-505	Introduction to Database Systems	3	0	4	5
3.	PCC CS-603	Machine Learning	3	1	0	4
4.	PCC CS-403	Operating Systems	3	0	4	5
5.	HSMC-501	Humanities II	3	0	0	3
6.	MC	Constitution of India/ Essence of Indian Knowledge Tradition	-	-	-	0
Total						**20**

Semester VI						
S.No	**Course Code**	**Course Title**	**L**	**T**	**P**	**Credits**
1.	PCC CS-601	Computer Networks	3	0	4	5
2.	PEC CS-601	Introductory Cyber Security	3	0	4	5
3.	PEC	Elective-I	3	0	0	3
4.	PEC	Elective-II	3	0	0	3
5.	PCC CS-504	Theory of Computation	3	1	0	4
6.	PROJ CS-601	Project-1	0	0	6	3
Total						**23**

Semester VII						
S.No	**Course Code**	**Course Title**	**L**	**T**	**P**	**Credits**
1.	PCC CS-602	Compiler Design	3	0	4	5
2.	PEC	Elective-III	3	0	0	3
3.	OEC	Open Elective-I	3	0	0	3
4.	BSC-701	Biology	2	1	0	3
5.	PROJ CS-701	Project-II	0	0	12	6
Total						**20**

Semester VIII						
S.No	**Course Code**	**Course Title**	**L**	**T**	**P**	**Credits**
1.	PEC	Elective-IV	3	0	0	3
2.	OEC	Open Elective-II	3	0	0	3
3.	OEC	Open Elective-III	3	0	0	3
4.	PROJ CS-801	Project-III	0	0	12	6
Total						**15**

SEMESTER – I

SEMESTER I

BSC-101	Physics-I	3L:1T:2P	5 Credits

Course Objective:

To enhance the fundamental knowledge in Physics and its applications relevant to various streams of Engineering and Technology.

1. Introduction to Electromagnetic Theory
Pre-requisites (if any): Mathematics course with vector calculus

Module I: Electrostatics in vacuum

Calculation of electric field and electrostatic potential for a charge distribution; Divergence and curl of electrostatic field; Laplace's and Poisson's equations for electrostatic potential and uniqueness of their solution and connection with steady state diffusion and thermal conduction; Practical examples like Faraday's cage and coffee-ring effect; Boundary conditions of electric field and electrostatic potential; method of images; energy of a charge distribution and its expression in terms of electric field.

Module II: Electrostatics in a linear dielectric medium

Electrostatic field and potential of a dipole. Bound charges due to electric polarization; Electric displacement; boundary conditions on displacement; Solving simple electrostatics problems in presence of dielectrics – Point charge at the center of a dielectric sphere, charge in front of a dielectric slab, dielectric slab and dielectric sphere in uniform electric field.

Module III: Magneto statics

Bio-Savart law, Divergence and curl of static magnetic field; vector potential and calculating it for a given magnetic field using Stokes' theorem; the equation for the vector potential and its solution for given current densities.

Module IV: Magneto statics in a linear magnetic medium

Magnetization and associated bound currents; auxiliary magnetic field H; Boundary conditions on B and H. Solving for magnetic field due to simple magnets like a bar magnet; magnetic susceptibility and ferromagnetic, paramagnetic and diamagnetic materials; Qualitative discussion of magnetic field in presence of magnetic materials.

Module V: Faraday's law

Faraday's law in terms of EMF produced by changing magnetic flux; equivalence of Faraday's law and motional EMF; Lenz's law; Electromagnetic breaking and its applications; Differential form of Faraday's law expressing curl of electric field in terms of time-derivative of magnetic field and

calculating electric field due to changing magnetic fields in quasi-static approximation; energy stored in a magnetic field.

Module VI: Displacement current, Magnetic field due to time-dependent electric field and Maxwell's equations

Continuity equation for current densities; Modifying equation for the curl of magnetic field to satisfy continuity equation; displace current and magnetic field arising from time dependent electric field; calculating magnetic field due to changing electric fields in quasistatic approximation. Maxwell's equation in vacuum and non-conducting medium; Energy in an electromagnetic field; Flow of energy and Pointing vector with examples. Qualitative discussion of momentum in electromagnetic fields.

Module VII: Electromagnetic waves

The wave equation; Plane electromagnetic waves in vacuum, their transverse nature and polarization; relation between electric and magnetic fields of an electromagnetic wave; energy carried by electromagnetic waves and examples. Momentum carried by electromagnetic waves and resultant pressure. Reflection and transmission of electromagnetic waves from a non-conducting medium-vacuum interface for normal incidence.

Laboratory/ Practicals:

Choice of experiments from the following:

1. Experiments on electromagnetic induction and electromagnetic braking;
2. LC circuit and LCR circuit;
3. Resonance phenomena in LCR circuits;
4. Magnetic field from Helmholtz coil;
5. Measurement of Lorentz force in a vacuum tube.

Text Books/Suggested References:

1. AICTE's Prescribed Textbook: Physics (Introduction to Electromagnetic Theory) with Lab Manual, Khanna Book Publishing Company.
2. Bhattacharya & Nag, Engineering Physics
3. David Griffiths, Introduction to Electrodynamics
4. Halliday and Resnick, Physics
5. W. Saslow, Electricity, magnetism and light
6. Malik, Singh, Engineering Physics, Tata McGraw Hill

Alternative NPTEL/SWAYAM Course:

S. No.	NPTEL Course Name	Instructor	Host Institute
1	INTRODUCTION TO ELECTROMAGNETIC THEORY	PROF. MANOJ HARBOLA	IIT KANPUR

EXPERIMENTS THAT MAY BE PERFORMED THROUGH VIRTUAL LABS:

S. No.	Experiment Name	Experiment Link(s)
1	LC circuit and LCR circuit;	1. http://vlab.amrita.edu/?sub=1&brch=75&sim=326&cnt=1 2. http://vlab.amrita.edu/?sub=1&brch=75&sim=330&cnt=1 3. http://vlab.amrita.edu/?sub=1&brch=75&sim=318&cnt=1 4. http://vlab.amrita.edu/?sub=1&brch=75&sim=325&cnt=1 5. http://vlabs.iitkgp.ernet.in/asnm/exp12/index.htm
2	Resonance phenomena in LCR circuits	http://vlab.amrita.edu/?sub=1&brch=75&sim=325&cnt=1

*****

2. Introduction to Mechanics
Pre-requisites (if any): High School Education

Module I

Transformation of scalars and vectors under Rotation transformation; Forces in Nature; Newton's laws and its completeness in describing particle motion; Form invariance of Newton's Second Law; Solving Newton's equations of motion in polar coordinates; Problems including constraints and friction; Extension to cylindrical and spherical coordinates.

Module II

Potential energy function; F = - Grad V, equipotential surfaces and meaning of gradient; Conservative and non-conservative forces, curl of a force field; Central forces; Conservation of Angular

Momentum; Energy equation and energy diagrams; Elliptical, parabolic and hyperbolic orbits; Kepler problem; Application: Satellite manoeuvres;

Module III

Non-inertial frames of reference; Rotating coordinate system: Five-term acceleration formula. Centripetal and Coriolis accelerations; Applications: Weather systems, Foucault pendulum;

Module IV

Harmonic oscillator; Damped harmonic motion – over-damped, critically damped and lightly-damped oscillators; Forced oscillations and resonance.

Module V

Definition and motion of a rigid body in the plane; Rotation in the plane; Kinematics in a coordinate system rotating and translating in the plane; Angular momentum about a point of a rigid body in planar motion; Euler's laws of motion, their independence from Newton's laws, and their necessity in describing rigid body motion; Examples.

Module VI

Introduction to three-dimensional rigid body motion — only need to highlight the distinction from two-dimensional motion in terms of (a) Angular velocity vector, and its rate of change and (b) Moment of inertia tensor; Three-dimensional motion of a rigid body wherein all points move in a coplanar manner: e.g. Rod executing conical motion with center of mass fixed — only need to show that this motion looks two-dimensional but is three-dimensional, and two-dimensional formulation fails.

Laboratory - Introduction to Mechanics

Suggested list of experiments from the following:

1. Coupled oscillators;
2. Experiments on an air-track;
3. Experiment on moment of inertia measurement,
4. Experiments with gyroscope;
5. Resonance phenomena in mechanical oscillators.

TEXTBOOKS/REFERENCES:

1. AICTE's Prescribed Textbook: Physics (Introduction to Mechanics) with Lab Manual, Khanna Book Publishing Co.
2. Engineering Physics, Bhattacharya & Nag
3. Engineering Mechanics, DS Bedi & MP Poonia
4. Engineering Mechanics, 2nd edition — MK Harbola
5. Introduction to Mechanics — MK Verma
6. An Introduction to Mechanics — D Kleppner & R Kolenkow
7. Principles of Mechanics — JL Synge & BA Griffiths

8. Mechanics — JP Den Hartog
9. Engineering Mechanics - Dynamics, 7th ed. - JL Meriam
10. Mechanical Vibrations — JP Den Hartog
11. Theory of Vibrations with Applications — WT Thomson

Alternative NPTEL/SWAYAM Course:

S. No.	NPTEL Course Name	Instructor	Host Institute
1	ENGINEERING MECHANICS	PROF. MANOJ HARBOLA	IIT KANPUR

EXPERIMENTS THAT MAY BE PERFORMED THROUGH VIRTUAL LABS:

S. No.	Experiment Name	Experiment Link(s)
1	Experiment on moment of inertia measurement.	https://vlab.amrita.edu/?sub=1&brch=74&sim=571&cnt=1

*****

3. Quantum Mechanics for Engineers
Pre-requisites (if any): Mathematics Course on Differential equations & linear algebra

Module I: Wave nature of particles and the Schrodinger equation

Introduction to Quantum mechanics, Wave nature of Particles, Time-dependent and time independent Schrodinger equation for wave function, born interpretation, probability current, Expectation values, Free-particle wave function and wave-packets, Uncertainty principle.

Module II: Mathematical Preliminaries for quantum mechanics

Complex numbers, Linear vector spaces, inner product, operators, eigenvalue problems, Hermitian operators, Hermite polynomials, Legendre's equation, spherical harmonics.

Module III: Applying the Schrodinger equation

Solution of stationary-state Schrodinger equation for one dimensional problems– particle in a box, particle in attractive delta-function potential, square-well potential, linear harmonic oscillator. Numerical solution of stationary-state Schrodinger equation for one dimensional problems for different potentials Scattering from a potential barrier and tunneling; related examples like alpha-decay, field ionization and scanning tunneling microscope Three-dimensional problems: particle in three dimensional box and related examples, Angular momentum operator, Rigid Rotor, Hydrogen

atom ground-state, orbitals, interaction with magnetic field, spin, Numerical solution stationary-state radial Schrodinger equation for spherically symmetric potentials.

Module IV: Introduction to molecular bonding

Particle in double delta-function potential, Molecules (hydrogen molecule, valence bond and molecular orbitals picture), singlet/triplet states, chemical bonding, hybridization.

Module V: Introduction to solids

Free electron theory of metals, Fermi level, density of states, Application to white dwarfs and neutron stars, Bloch's theorem for particles in a periodic potential, Kronig-Penney model and origin of energy bands Numerical solution for energy in one-dimensional periodic lattice by mixing plane waves.

Laboratory - Quantum Mechanics for Engineers

Suggested list of experiments: Frank-Hertz experiment; photoelectric effect experiment; recording hydrogen atom spectrum.

TEXTBOOKS/REFERENCES:

1. AICTE's Prescribed Textbook: Physics (Quantum Mechanics for Engineers) with Lab Manual, Khanna Book Publishing Co.
2. Bhattacharya & Nag, Engineering Physics
3. Eisberg and Resnick, Introduction to Quantum Physics
4. D. J. Griffiths, Quantum Mechanics
5. Richard Robinett, Quantum Mechanics
6. Daniel McQuarrie, Quantum Chemistry

Alternative NPTEL/SWAYAM Course:

S. No.	NPTEL Course Name	Instructor	Host Institute
1	INTRODUCTION TO ELECTROMAGNETIC THEORY	PROF. MANOJ HARBOLA	IIT KANPUR
2	QUANTUM MECHANICS I	PROF. P. RAMADEVI	IIT BOMBAY

EXPERIMENTS THAT MAY BE PERFORMED THROUGH VIRTUAL LABS:

S. No.	Experiment Name	Experiment Link(s)
1	Photoelectric effect experiment.	http://mpv-au.vlabs.ac.in/modern-physics/Photo_Electric_Effect/

*****

4. Oscillations, waves and optics
Pre-requisites (if any): Mathematics Course on Differential equations

Module I: Simple harmonic motion, damped and forced simple harmonic oscillator

Mechanical and electrical simple harmonic oscillators, complex number notation and phasor representation of simple harmonic motion, damped harmonic oscillator – heavy, critical and light damping, energy decay in a damped harmonic oscillator, quality factor, forced mechanical and electrical oscillators, electrical and mechanical impedance, steady state motion of forced damped harmonic oscillator, power absorbed by oscillator.

Module II: Non-dispersive transverse and longitudinal waves in one dimension and introduction to dispersion

Transverse wave on a string, the wave equation on a string, Harmonic waves, reflection and transmission of waves at a boundary, impedance matching, standing waves and their Eigen frequencies, longitudinal waves and the wave equation for them, acoustics waves and speed of sound, standing sound waves. Waves with dispersion, water waves, superposition of waves and Fourier method, wave groups and group velocity.

Module III: The propagation of light and geometric optics

Fermat's principle of stationary time and its applications e.g. in explaining mirage effect, laws of reflection and refraction, Light as an electromagnetic wave and Fresnel equations, reflectance and transmittance, Brewster's angle, total internal reflection, and evanescent wave. Mirrors and lenses and optical instruments based on them, transfer formula and the matrix method.

Module IV: Wave optics

Huygens' principle, superposition of waves and interference of light by wave front splitting and amplitude splitting; Young's double slit experiment, Newton's rings, Michelson interferometer, Mach-Zehnder interferometer.

Farunhofer diffraction from a single slit and a circular aperture, the Rayleigh criterion for limit of resolution and its application to vision; Diffraction gratings and their resolving power.

Module V: Lasers

Einstein's theory of matter radiation interaction and A and B coefficients; amplification of light by population inversion, different types of lasers: gas lasers (He-Ne, CO2), solid-state lasers (ruby, Neodymium), dye lasers; Properties of laser beams: mono-chromaticity, coherence, directionality and brightness, laser speckles, applications of lasers in science, engineering and medicine.

Laboratory - Oscillations, waves and optics

Suggested list of experiments from the following:

- Diffraction and interference experiments (from ordinary light or laser pointers); measurement of speed of light on a table top using modulation; minimum deviation from a prism.

TEXTBOOKS/REFERENCES:

1. AICTE's Prescribed Textbook: Physics (Oscillations, Waves & Optics) with Lab Manual, Khanna Book Publishing.
2. Bhattacharya & Nag, Engineering Physics
3. Ian G. Main, Oscillations and waves in physics
4. H.J. Pain, The physics of vibrations and waves
5. E. Hecht, Optics
6. A. Ghatak, Optics
7. O. Svelto, Principles of Lasers

Alternative NPTEL/SWAYAM Course:

S. No.	NPTEL Course Name	Instructor	Host Institute
1	WAVES AND OSCILLATIONS	PROF. M. S. SANTHANAM	IISER PUNE

EXPERIMENTS THAT MAY BE PERFORMED THROUGH VIRTUAL LABS:

S. No.	Experiment Name	Experiment Link(s)
1	Diffraction and interference experiments (from ordinary light or laser pointers).	http://ov-au.vlabs.ac.in/optics/Diffraction_Grating/
2	Minimum deviation from a prism.	http://ov-au.vlabs.ac.in/optics/Spectrometer_i_d_Curve/

******

BS-102	Mathematics-I	3L:1T:0P	4 Credits

Course Objectives:

The goal of this course is to achieve conceptual understanding and to retain the best traditions of traditional calculus. The syllabus is designed to provide the basic tools of calculus mainly for the purpose of modelling the engineering problems mathematically and obtaining solutions. This is a foundation course which mainly deals with topics such as single variable and multivariable calculus and plays an important role in the understanding of science, engineering, economics and computer science, among other disciplines.

Course Contents:

Module 1: Basic Calculus: (6 hours)

Curvature, evolutes and involutes; Evaluation of definite and improper integrals; Beta and Gamma functions and their properties; Applications of definite integrals to evaluate surface areas and volumes of revolutions.

Module 2: Single-variable Calculus (Differentiation): (6 hours)

Rolle's Theorem, Mean value theorems and applications; Extreme values of functions; Linear approximation; Indeterminate forms and L' Hospital's rule.

Module 3: Sequences and series: (10 hours)

Limits of sequence of numbers, Calculation of limits, Infinite series; Tests for convergence; Power series, Taylor and Maclaurin series; Taylor theorem, convergence of Taylor series, error estimates.

Module 4: Multivariable Calculus (Differentiation): (8 hours)

Limit, continuity and partial derivatives, directional derivatives, gradient, total derivative; Tangent plane and normal line; Maxima, minima and saddle points; Method of Lagrange multipliers.

Module 5: **Multivariable Calculus (Integration):** (10 hours)

Multiple Integration: Double integrals (Cartesian), change of order of integration in double integrals, Change of variables (Cartesian to polar), Applications: areas and volumes, Center of mass and Gravity (constant and variable densities); Triple integrals (Cartesian), orthogonal curvilinear coordinates, Simple applications involving cubes, sphere and rectangular parallelepipeds; Scalar line integrals, vector line integrals, scalar surface integrals, vector surface integrals, Gradient, curl and divergence, Theorems of Green, Gauss and Stokes.

TEXTBOOKS/REFERENCES:

1. AICTE's Prescribed Textbook: Mathematics-I (Calculus & Linear Algebra), Khanna Book Publishing Co.
2. Reena Garg, Engineering Mathematics, Khanna Book Publishing Company, 2022.
3. Reena Garg, Advanced Engineering Mathematics, Khanna Book Publishing Company, 2021.
4. G.B. Thomas and R.L. Finney, Calculus and Analytic geometry, 9th Edition, Pearson, Reprint, 2002.
5. Erwin Kreyszig, Advanced Engineering Mathematics, 9th Edition, John Wiley & Sons, 2006.
6. Ramana B.V., Higher Engineering Mathematics, Tata McGraw Hill New Delhi, 11th Reprint, 2010.
7. Veerarajan T., Engineering Mathematics for first year, Tata McGraw-Hill, New Delhi, 2008.
8. N.P. Bali and Manish Goyal, A text book of Engineering Mathematics, Laxmi Publications, Reprint, 2008.
9. B.S. Grewal, Higher Engineering Mathematics, Khanna Publishers, 36th Edition, 2010.

Note: The modules have been prepared keeping the following from the Textbooks/References in mind:

(1) Module 1: The relevant sections from Chapters 2, 6 and 11 of [3].
(2) Module 2: Sections 3.1, 3.2, 3.3, 3.7 & 6.6 of [1].
(3) Module 3: Sections 8.1-8.6, 8.8-8.10 of [1].
(4) Module 4: Sections 12.1-12.5, 12.7-12.9 of [1].
(5) Module 5: Sections 13.1 – 13.7, 14.1 – 14.8 of [1].

Course outcomes: The objective of this course is to familiarize the prospective engineers with techniques in calculus, multivariate differentiation and integration. It aims to equip the students with standard concepts and tools at an intermediate to advanced level that will serve them well towards tackling more advanced level of mathematics and applications that they would find useful in their disciplines.

The students will learn

- To apply differential and integral calculus to notions of curvature and to improper integrals. Apart from some other applications they will have a basic understanding of Beta and Gamma functions.
- The fallouts of Rolle's Theorem that is fundamental to application of analysis to Engineering problems.
- The tool of power series and Fourier series for learning advanced Engineering Mathematics.
- To deal with functions of several variables that are essential in most branches of engineering.
- To acquaint the student with mathematical tools needed in evaluating multiple integrals and their usage.

******

ESC-101	Basic Electrical Engineering	3L:1T:2P	5 Credits

Course Objective:

The objective of this Course is to provide the students with an introductory and broad treatment of the field of Electrical Engineering.

Course Contents:

Module I: D. C. Circuits covering, Ohm's Law and Kirchhoff's Laws; Analysis of series, parallel and series-parallel circuits excited by independent voltage sources; Power and energy; Electromagnetism covering, Faradays Laws, Lenz's Law, Fleming's Rules, Statically and dynamically induced EMF; Concepts of self-inductance, mutual inductance and coefficient of coupling; Energy stored in magnetic fields;

Module II: Single Phase A.C. Circuits covering, Generation of sinusoidal voltage- definition of average value, root mean square value, form factor and peak factor of sinusoidal voltage and current and phasor representation of alternating quantities; Analysis with phasor diagrams of R, L, C, RL, RC and RLC circuits; Real power, reactive power, apparent power and power factor, series, parallel and series- parallel circuits; Three Phase A.C. Circuits covering, Necessity and Advantages of three phase systems, Generation of three phase power, definition of Phase sequence, balanced supply and balanced load; Relationship between line and phase values of balanced star and delta connections; Power in balanced three phase circuits, measurement of power by two wattmeter method;

Module III: Transformers covering, Principle of operation and construction of single phase transformers (core and shell types). EMF equation, losses, efficiency and voltage regulation; Synchronous Generators covering, Principle of operation; Types and constructional features; EMF equation;

Module IV: DC Machines covering, working principle of DC machine as a generator and a motor; Types and constructional features; EMF equation of generator, relation between EMF induced and terminal voltage enumerating the brush drop and drop due to armature reaction; DC motor working principle; Back EMF and its significance, torque equation; Types of D.C. motors, characteristics and applications; Necessity of a starter for DC motor;

Module V: Three Phase Induction Motors covering; Concept of rotating magnetic field; Principle of operation, types and constructional features; Slip and its significance; Applications of squirrel cage and slip ring motors; Necessity of a starter, star-delta starter.

Module VI: Sources of Electrical Power covering, Introduction to Wind, Solar, Fuel cell, Tidal, Geo-thermal, Hydroelectric, Thermal-steam, diesel, gas, nuclear power plants; Concept of cogeneration, and distributed generation;

TEXT/REFERENCS BOOKS:

1. AICTE's Prescribed Textbook: Basic Electrical Engineering, Khanna Book Publishing.

2. Ritu Sahdev (2022), Basic Electrical Engineering, Khanna Book Publishing.
3. Nagrath I.J. and D. P. Kothari (2001), Basic Electrical Engineering, Tata McGraw Hill.
4. Hayt and Kimberly, Engineering Circuit Analysis, Tata McGraw Hill.
5. Kulshreshtha D.C. (2009), Basic Electrical Engineering, Tata McGraw Hill.
6. Rajendra Prasad (2009), Fundamentals of Electrical Engineering, Prentice Hall, India Hughes, E. 2005)
7. Mittel & Mittal, Basic Electrical Engineering, Tata McGraw Hill.

Alternative NPTEL/SWAYAM Course:

S. No.	NPTEL Course Name	Instructor	Host Institute
1	BASIC ELECTRIC CIRCUITS	PROF. ANKUSH SHARMA	IIT KANPUR
2	BASIC ELECTRICAL CIRCUITS	PROF. NAGENDRA KRISHNAPURA	IITM
3	FUNDAMENTALS OF ELECTRICAL ENGINEERING	PROF. DEBAPRIYA DAS	IIT KGP

COURSE OUTCOMES:

The students will learn:

1. To explain strong basics of Electrical Engineering and practical implementation of Electrical fundamentals.
2. To identify different applications of commonly used electrical machinery.

******

ESC-102	Engineering Graphics & Design	1L:0T:4P	3 Credits

COURSE OBJECTIVE(S):

The objective of this Course is to provide the basic knowledge about Engineering Drawing. Detailed concepts are given in projections, technical drawing, dimensioning and specifications, so useful for a student in preparing for an engineering career.

COURSE CONTENTS:

Traditional Engineering Graphics: Principles of Engineering Graphics; Orthographic Projection; Descriptive Geometry; Drawing Principles; Isometric Projection; Surface Development; Perspective; Reading a Drawing; Sectional Views; Dimensioning & Tolerances; True Length, Angle; intersection, Shortest Distance.

Computer Graphics: Engineering Graphics Software; -Spatial Transformations; Orthographic Projections; Model Viewing; Co-ordinate Systems; Multi-view Projection; Exploded Assembly; Model Viewing; Animation; Spatial Manipulation; Surface Modelling; Solid Modelling; Introduction to Building Information Modelling (BIM).

(Except the basic essential concepts, most of the teaching part can happen concurrently in the laboratory)

Module I: Introduction to Engineering Drawing

Principles of Engineering Graphics and their significance, usage of Drawing instruments, lettering, Conic sections including the Rectangular Hyperbola (General method only); Cycloid, Epicycloid, Hypocycloid and Involute; Scales – Plain, Diagonal and Vernier Scales;

Module II: Orthographic Projections

Principles of Orthographic Projections-Conventions - Projections of Points and lines inclined to both planes; Projections of planes inclined Planes - Auxiliary Planes;

Module III: Projections of Regular Solids

Covering those inclined to both the Planes- Auxiliary Views; Draw simple annotation, dimensioning and scale. Floor plans that include: windows, doors, and fixtures such as WC, bath, sink, shower, etc.

Module IV: Sections and Sectional Views of Right Angular Solids

Prism, Cylinder, Pyramid, Cone – Auxiliary Views; Development of surfaces of Right Regular Solids - Prism, Pyramid, Cylinder and Cone; Draw the sectional orthographic views of geometrical solids, objects from industry and dwellings (foundation to slab only).

Module V: Isometric Projections

Principles of Isometric projection – Isometric Scale, Isometric Views, Conventions; Isometric Views of lines, Planes, Simple and compound Solids; Conversion of Isometric Views to Orthographic Views and Vice-versa, Conventions;

Module VI: Overview of Computer Graphics

Listing the computer technologies that impact on graphical communication, Demonstrating knowledge of the theory of CAD software [such as: The Menu System, Toolbars (Standard, Object Properties, Draw, Modify and Dimension), Drawing Area (Background, Crosshairs, Coordinate System), Dialog boxes and windows, Shortcut menus (Button Bars), The Command Line (where applicable), The Status Bar, Different methods of zoom as used in CAD, Select and erase objects.; Isometric Views of lines, Planes, Simple and compound Solids];

Module VII: Customisation & CAD Drawing

Consisting of set up of the drawing page and the printer, including scale settings, setting up of Modules and drawing limits; ISO and ANSI standards for coordinate dimensioning and tolerancing; Orthographic constraints, Snap to objects manually and automatically; Producing drawings by using various coordinate input entry methods to draw straight lines, Applying various ways of drawing circles;

Module VIII: Annotations, layering & other functions

Covering applying dimensions to objects, applying annotations to drawings; Setting up and use of Layers, layers to create drawings, Create, edit and use customized layers; Changing line lengths through modifying existing lines (extend/lengthen); Printing documents to paper using the print command; orthographic projection techniques; Drawing sectional views of composite right regular geometric solids and project the true shape of the sectioned surface; Drawing annotation, Computer-aided design (CAD) software modeling of parts and assemblies. Parametric and non-parametric solid, surface, and wireframe models. Part editing and two-dimensional documentation of models. Planar projection theory, including sketching of perspective, isometric, multiview, auxiliary, and section views. Spatial visualization exercises. Dimensioning guidelines, tolerancing techniques; dimensioning and scale multi views of dwelling;

Module IX: Demonstration of a simple team design project that illustrates

Geometry and topology of engineered components: creation of engineering models and their presentation in standard 2D blueprint form and as 3D wire-frame and shaded solids; meshed topologies for engineering analysis and tool-path generation for component manufacture; geometric dimensioning and tolerancing; Use of solid-modeling software for creating associative models at the component and assembly levels; floor plans that include: windows, doors, and fixtures such as WC, bath, sink, shower, etc. Applying colour coding according to building drawing practice; Drawing sectional elevation showing foundation to ceiling; Introduction to Building Information Modelling (BIM).

Text/Reference Books:

1. AICTE's Prescribed Textbook: Engineering Graphics & Design Khanna Book Publishing.
2. Jain, Maheshwari, Gautam (2021), Engineering Graphics & Design, Khanna Book Publishing.
3. Bhatt N.D., Panchal V.M. & Ingle P.R., (2014), Engineering Drawing, Charotar Publishing.
4. Shah, M.B. & Rana B.C. (2008), Engineering Drawing and Computer Graphics, Pearson.
5. Agrawal B. & Agrawal C. M. (2012), Engineering Graphics, TMH Publication.
6. Narayana, K.L. & P Kannaiah (2008), Text book on Engineering Drawing, Scitech Publishers.
7. (Corresponding set of) CAD Software Theory and User Manuals.

Alternative NPTEL/SWAYAM Course:

S. No.	NPTEL Course Name	Instructor	Host Institute
1	PROF. RAJARAM LAKKARAJU	IIT KHARAGPUR	ENGINEERING DRAWING AND COMPUTER GRAPHICS
2	PROF. NIHAR RANJAN PATRA	IIT KANPUR	ENGINEERING GRAPHICS

Course Outcomes:

All phases of manufacturing or construction require the conversion of new ideas and design concepts into the basic line language of graphics. Therefore, there are many areas (civil, mechanical, electrical, architectural and industrial) in which the skills of the CAD technicians play major roles in the design and development of new products or construction. Students prepare for actual work situations through practical training in a new state-of-the-art computer designed CAD laboratory using engineering software. This course is designed to address:

- to prepare you to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability
- to prepare you to communicate effectively
- to prepare you to use the techniques, skills, and modern engineering tools necessary for engineering practice

The students will learn:

- To describe engineering design and its place in society.
- To discuss the visual aspects of engineering design.
- To use engineering graphics standards.
- To illustrate solid modelling.
- To use computer-aided geometric design.
- To design creating working drawings.
- To inspect engineering communication.

****

HSMC-102	Design Thinking	0L:0T:2P	1 Credit

COURSE OBJECTIVE(S):

The objective of this Course is to provide the new ways of creative thinking and Learn the innovation cycle of Design Thinking process for developing innovative products which useful for a student in preparing for an engineering career.

COURSE CONTENTS:

Unit 1: An Insight to Learning
Understanding the Learning Process, Kolb's Learning Styles, Assessing and Interpreting

Unit 2: Remembering Memory
Understanding the Memory process, Problems in retention, Memory enhancement techniques

Unit 3: Emotions: Experience & Expression
Understanding Emotions: Experience & Expression, Assessing Empathy, Application with Peers

Unit 4: Basics of Design Thinking
Definition of Design Thinking, Need for Design Thinking, Objective of Design Thinking, Concepts & Brainstorming, Stages of Design Thinking Process (explain with examples) – Empathize, Define, Ideate, Prototype, Test

Unit 5: Being Ingenious & Fixing Problem
Understanding Creative thinking process, Understanding Problem Solving, Testing Creative Problem Solving

Unit 6: Process of Product Design
Process of Engineering Product Design, Design Thinking Approach, Stages of Product Design, Examples of best product designs and functions, Assignment – Engineering Product Design

Unit 7: Prototyping & Testing
What is Prototype? Why Prototype? Rapid Prototype Development process, Testing, **Sample Example**, Test Group Marketing

Unit 8: Celebrating the Difference
Understanding Individual differences & Uniqueness, Group Discussion and Activities to encourage the understanding, acceptance and appreciation of Individual differences

Unit 9: Design Thinking & Customer Centricity

Practical Examples of Customer Challenges, Use of Design Thinking to Enhance Customer Experience, Parameters of Product experience, Alignment of Customer Expectations with Product Design

Unit 10: Feedback, Re-Design & Re-Create

Feedback loop, Focus on User Experience, Address "ergonomic challenges, User focused design, rapid prototyping & testing, final product, Final Presentation – **"Solving Practical Engineering Problem through Innovative Product Design & Creative Solution".**

Text/Reference Books:

1. E Balaguruswamy (2022), Developing Thinking Skills (The way to Success), Khanna Book Publishing Company.

Course Outcomes (CO):

Student will able to:

Compare and classify the various learning styles and memory techniques and Apply them in their engineering education

Analyze emotional experience and Inspect emotional expressions to better understand users while designing innovative products

Develop new ways of creative thinking and Learn the innovation cycle of Design Thinking process for developing innovative products

Propose real-time innovative engineering product designs and Choose appropriate frameworks, strategies, techniques during prototype development

Perceive individual differences and its impact on everyday decisions and further Create a better customer experience

*****

AU-101	IDEA Lab Workshop	2L:0T:4P	0 Credit

Course Objectives:

1. To learn all the skills associated with the tools and inventory associated with the IDEA Lab.
2. Learn useful mechanical and electronic fabrication processes.
3. Learn necessary skills to build useful and standalone system/ project with enclosures.
4. Learn necessary skills to create print and electronic documentation for the system/project

Course Contents:

Unit #	Topics	
1.	Electronic component familiarization, Understanding electronic system design flow. Schematic design and PCB layout and Gerber creation using EagleCAD. Documentation using Doxygen, Google Docs, Overleaf. Version control tools - GIT and GitHub. Basic 2D and 3D designing using CAD tools such as FreeCAD, Sketchup, Prusa Slicer, FlatCAM, Inkspace, OpenBSP and VeriCUT.	Introduction to basic hand tools - Tape measure, combination square, Vernier caliper, hammers, fasteners, wrenches, pliers, saws, tube cutter, chisels, vice and clamps, tapping and threading. Adhesives Introduction to Power tools: Power saws, band saw, jigsaw, angle grinder, belt sander, bench grinder, rotary tools. Various types of drill bits,
2.	Familiarization and use of basic measurement instruments - DSO including various triggering modes, DSO probes, DMM, LCR bridge, Signal and function generator. Logic analyzer and MSO. Bench power supply (with 4-wire output) Circuit prototyping using (a) breadboard, (b) Zero PCB (c) 'Manhattan' style and (d) custom PCB. Single, double and multilayer PCBs. Single and double-sided PCB prototype fabrication in the lab. Soldering using soldering iron/station. Soldering using a temperature controlled reflow oven. Automated circuit assembly and soldering using pick and place machines.	Mechanical cutting processes - 3-axis CNC routing, basic turning, milling, drilling and grinding operations, Laser cutting, Laser engraving etc. Basic welding and brazing and other joining techniques for assembly. Concept of Lab aboard a Box.

3.	Electronic circuit building blocks including common sensors. Arduino and Raspberry Pi programming and use. Digital Input and output. Measuring time and events. PWM. Serial communication. Analog input. Interrupts programming. Power Supply design (Linear and Switching types), Wireless power supply, USB PD, Solar panels, Battery types and charging	3D printing and prototyping technology – 3D printing using FDM, SLS and SLA. Basics of 3D scanning, point cloud data generation for reverse engineering. Prototyping using subtractive cutting processes. 2D and 3D Structures for prototype building using Laser cutter and CNC routers. Basics of IPR and patents; Accessing and utilizing patent information in IDEA Lab
4.	Discussion and implementation of a mini project.	
5.	Documentation of the mini project (Report and video).	

Laboratory Activities:

S. No.	List of Lab activities and experiments
1.	Schematic and PCB layout design of a suitable circuit, fabrication and testing of the circuit.
2.	Machining of 3D geometry on soft material such as soft wood or modelling wax.
3.	3D scanning of computer mouse geometry surface. 3D printing of scanned geometry using FDM or SLA printer.
4.	2D profile cutting of press fit box/casing in acrylic (3 or 6 mm thickness)/cardboard, MDF (2 mm) board using laser cutter & engraver.
5.	2D profile cutting on plywood /MDF (6-12 mm) for press fit designs.
6.	Familiarity and use of welding equipment.
7.	Familiarity and use of normal and wood lathe.
8.	Embedded programming using Arduino and/or Raspberry Pi.
9.	Design and implementation of a capstone project involving embedded hardware, software and machined or 3D printed enclosure.

Reference Books:

S. No.	Title
1.	AICTE's Prescribed Textbook: Workshop / Manufacturing Practices (with Lab Manual), Khanna Book Publishing.
2.	All-in-One Electronics Simplified, A.K. Maini; 2021. ISBN-13: 978-9386173393, Khanna Book Publishing Company, New Delhi.
3.	Simplified Q&A - Data Science with Artificial Intelligence, Machine Learning and Deep Learning, Rajiv Chopra, ISBN: 978-9355380821, Khanna Book Publishing Company, New Delhi.
4.	3D Printing & Design, Dr. Sabrie Soloman, ISBN: 978-9386173768, Khanna Book Publishing Company, New Delhi.
5.	The Big Book of Maker Skills: Tools & Techniques for Building Great Tech Projects. Chris Hackett. Weldon Owen; 2018. ISBN-13: 978-1681884325.
6.	The Total Inventors Manual (Popular Science): Transform Your Idea into a Top-Selling Product. Sean Michael Ragan (Author). Weldon Owen; 2017. ISBN-13: 978-1681881584.
7.	Make: Tools: How They Work and How to Use Them. Platt, Charles. Shroff/Maker Media. 2018. ISBN-13: 978-9352137374
8.	The Art of Electronics. 3rd edition. Paul Horowitz and Winfield Hill. Cambridge University Press. ISBN: 9780521809269
9.	Practical Electronics for Inventors. 4th edition. Paul Sherz and Simon Monk. McGraw Hill. ISBN-13: 978-1259587542
10.	Encyclopedia of Electronic Components (Volume 1, 2 and 3). Charles Platt. Shroff Publishers. ISBN-13: 978-9352131945, 978-9352131952, 978-9352133703
11.	Building Scientific Apparatus. 4th edition. John H. Moore, Christopher C. Davis, Michael A. Coplan and Sandra C. Greer. Cambridge University Press. ISBN-13: 978-0521878586
12.	Programming Arduino: Getting Started with Sketches. 2nd edition. Simon Monk. McGraw Hill. ISBN-13: 978-1259641633
13.	Make Your Own PCBs with EAGLE: From Schematic Designs to Finished Boards. Simon Monk and Duncan Amos. McGraw Hill Education. ISBN-13 : 978-1260019193.
14.	Pro GIT. 2nd edition. Scott Chacon and Ben Straub. A press. ISBN-13 : 978-1484200773
15.	Venuvinod, PK., MA. W., Rapid Prototyping – Laser Based and Other Technologies, Kluwer.
16.	Ian Gibson, David W Rosen, Brent Stucker., "Additive Manufacturing Technologies: Rapid Prototyping to Direct Digital Manufacturing", Springer, 2010
17.	Chapman W.A.J, "Workshop Technology", Volume I, II, III, CBS Publishers and distributors, 5th Edition,2002.

******

SEMESTER – II

SEMESTER II

BSC-202	Chemistry- I	3L:1T:2P	5 Credits

Course Objective:

The objective of the Chemistry-I is to acquaint the students with the basic phenomenon/concepts of chemistry, the student faces during course of their study in the industry and Engineering field. The student with the knowledge of the basic chemistry, will understand and explain scientifically the various chemistry related problems in the industry/engineering field. The student will able to understand the new developments and breakthroughs efficiently in engineering and technology. The introduction of the latest (R&D oriented) topics will make the engineering student upgraded with the new technologies.

Course Content:

Module I: Atomic and Molecular Structure

Schrodinger equation. Particle in a box solutions and their applications for conjugated molecules and nanoparticles. Forms of the hydrogen atom wave functions and the plots of these functions to explore their spatial variations. Molecular orbitals of diatomic molecules and plots of the multicenter orbitals. Equations for atomic and molecular orbitals. Energy level diagrams of diatomic. Pi-molecular orbitals of butadiene and benzene and aromaticity. Crystal field theory and the energy level diagrams for transition metal ions and their magnetic properties. Band structure of solids and the role of doping on band structures.

Module II: Spectroscopic techniques and applications

Principles of spectroscopy and selection rules. Electronic spectroscopy. Fluorescence and its applications in medicine. Vibrational and rotational spectroscopy of diatomic molecules. Applications. Nuclear magnetic resonance and magnetic resonance imaging, surface characterization techniques. Diffraction and scattering.

Module III: Intermolecular forces and potential energy surfaces

Ionic, dipolar and van Der Waals interactions. Equations of state of real gases and critical phenomena. Potential energy surfaces of H3, H2F and HCN and trajectories on these surfaces.

Module IV: Use of free energy in chemical equilibria (6 lectures)

Thermodynamic functions: energy, entropy and free energy. Estimations of entropy and free energies. Free energy and EMF. Cell potentials, the Nernst equation and applications. Acid base, oxidation reduction and solubility equilibria. Water chemistry. Corrosion. Use of free energy considerations in metallurgy through Ellingham diagrams.

Module V: Periodic properties

Effective nuclear charge, penetration of orbitals, variations of s, p, d and f orbital energies of atoms in the periodic table, electronic configurations, atomic and ionic sizes, ionization energies, electron affinity and electronegativity, polarizability, oxidation states, coordination numbers and geometries, hard soft acids and bases, molecular geometries.

Module VI: Stereochemistry

Representations of 3 dimensional structures, structural isomers and stereoisomers, configurations and symmetry and chirality, enantiomers, diastereomers, optical activity, absolute configurations and conformational analysis. Isomerism in transitional metal compounds.

Module VII: Organic reactions and synthesis of a drug molecule

Introduction to reactions involving substitution, addition, elimination, oxidation, reduction, cyclization and ring openings. Synthesis of a commonly used drug molecule.

LABORATORY

Choice of 10-12 experiments from the following:

1. Determination of surface tension and viscosity.
2. Thin layer chromatography.
3. Ion exchange column for removal of hardness of water.
4. Determination of chloride content of water.
5. Colligative properties using freezing point depression.
6. Determination of the rate constant of a reaction.
7. Determination of cell constant and conductance of solutions.
8. Potentiometry - determination of redox potentials and EMFs.
9. Synthesis of a polymer/drug.
10. Saponification/acid value of an oil.
11. Chemical analysis of a salt.
12. Lattice structures and packing of spheres.
13. Models of potential energy surfaces.
14. Chemical oscillations- Iodine clock reaction.
15. Determination of the partition coefficient of a substance between two immiscible liquids.
16. Adsorption of acetic acid by charcoal.
17. Use of the capillary viscosimeters to the demonstrate of the isoelectric point as the pH of minimum viscosity for gelatin sols and/or coagulation of the white part of egg.

Text/Reference Books:

1. AICTE's Prescribed Textbook: Chemistry – I with Lab Manual, Khanna Book Publishing.
2. Engineering Chemistry, by Manisha Agrawal.
3. University chemistry, by B. H. Mahan.
4. Chemistry: Principles and Applications, by M. J. Sienko and R. A. Plane.
5. Fundamentals of Molecular Spectroscopy, by C. N. Banwell.
6. Engineering Chemistry (NPTEL Web-book), by B. L. Tembe, Kamaluddin and M. S. Krishnan.

7. Physical Chemistry, by P. W. Atkins.
8. A Textbook of Engineering Chemistry, Shashi Chawla.
9. Organic Chemistry: Structure and Function by K. P. C. Volhardt and N. E. Schore, 5th Edition http://bcs.whfreeman.com/vollhardtschore5e/default.asp

\Alternative NPTEL/SWAYAM Course:

S. No.	NPTEL Course Name	Instructor	Host Institute
1	CHEMISTRY - I	PROF. MANGALA SUNDER KRISHNAN	IITM

EXPERIMENTS THAT MAY BE PERFORMED THROUGH VIRTUAL LABS:

S. No.	Experiment Name	Experiment Link(s)
1	Determination of surface tension and viscosity.	http://pcv-au.vlabs.ac.in/physical-chemistry/Determination_of_Viscosity_of_Organic_Solvents/
2	Ion exchange column for removal of hardness of water.	http://icv-au.vlabs.ac.in/inorganic-chemistry/Water_Analysis_Determination_of_Chemical_Parameters/
3	Determination of chloride content of water.	http://vlabs.iitb.ac.in/vlabs-dev/labs/nitk_labs/Environmental_Engineering_1/experiments/determination-of-chloride-nitk/simulation.html
4	Colligative properties using freezing point depression.	http://pcv-au.vlabs.ac.in/physical-chemistry/Cryoscopy/
5	Determination of the rate constant of a reaction.	http://pcv-au.vlabs.ac.in/physical-chemistry/EMF_Measurement/
6	Determination of cell constant and conductance of solutions.	http://icv-au.vlabs.ac.in/inorganic-chemistry/Water_Analysis_Determination_of_Physical_Parameters/

7	Potentiometry - determination of redox potentials and EMFs.	http://pcv-au.vlabs.ac.in/physical-chemistry/EMF_Measurement/
8	Saponification/acid value of an oil.	http://biotech01.vlabs.ac.in/bio-chemistry/Estimation_of_Saponification_Value_of_Fats_or_Oils/
9	Lattice structures and packing of spheres.	https://vlab.amrita.edu/?sub=1&brch=282&sim=370&cnt=1

Course Outcomes: The concepts developed in this course will aid in quantification of several concepts in chemistry that have been introduced at the 10+2 levels in schools. Technology is being increasingly based on the electronic, atomic and molecular level modifications. Quantum theory is more than 100 years old and to understand phenomena at nanometer levels, one has to base the description of all chemical processes at molecular levels. The course will enable the students:

- To analyse microscopic chemistry in terms of atomic and molecular orbitals and intermolecular forces.
- To rationalise bulk properties and processes using thermodynamic considerations.
- To distinguish the ranges of the electromagnetic spectrum used for exciting different molecular energy levels in various spectroscopic techniques
- To rationalise periodic properties such as ionization potential, electronegativity, oxidation states and electronegativity.
- To list major chemical reactions that are used in the synthesis of molecules.

Laboratory Outcomes: The chemistry laboratory course will consist of experiments illustrating the principles of chemistry relevant to the study of science and engineering. The students will learn:

- To estimate rate constants of reactions from concentration of reactants/products as a function of time.
- To measure molecular/system properties such as surface tension, viscosity, conductance of solutions, redox potentials, chloride content of water, etc.
- To synthesize a small drug molecule and analyze a salt sample.

******

BSC-201	Mathematics- II	3L:1T:0P	4 Credits

Course Objective:

Mathematics fundamental necessary to formulate, solve and analyse engineering problems.

Course Content:

Module 1: Matrices (10 hours)

Linear Systems of Equations; Linear Independence; Rank of a Matrix; Determinant, Inverse of a matrix, rank-nullity theorem; System of linear equations; Symmetric, skew-symmetric and orthogonal matrices; Determinants; Eigenvalues and eigenvectors; Orthogonal transformation; Diagonalization of matrices; Cayley-Hamilton Theorem.

Module 2: First order ordinary differential equations: (6 hours)

Exact, linear and Bernoulli's equations. Equations not of first degree: equations solvable for p, equations solvable for y, equations solvable for x and Clairaut's type.

Module 3: Ordinary differential equations of higher orders: (8 hours)

Second order linear differential equations with variable coefficients: Euler-Cauchy equations, solution by variation of parameters; Power series solutions: Legendre's equations and Legendre polynomials, Frobenius method, Bessel's equation and Bessel's functions of the first kind and their properties.

Module 4: Complex Variable – Differentiation: (8 hours):

Differentiation, Cauchy-Riemann equations, analytic functions, harmonic functions, finding harmonic conjugate; elementary analytic functions (exponential, trigonometric, logarithm) and their properties; Conformal mappings, Mobius transformations and their properties.

Module 5: Complex Variable – Integration: (8 hours):

Contour integrals, Cauchy-Goursat theorem (without proof), Cauchy Integral formula (without proof), Liouville's theorem and Maximum-Modulus theorem (without proof); Taylor's series, zeros of analytic functions, singularities, Laurent's series; Residues, Cauchy Residue theorem (without proof), Evaluation of definite integral involving sine and cosine, Evaluation of certain improper integrals using the Bromwich contour.

TEXTBOOKS/REFERENCES:

1. AICTE's Prescribed Textbook: Mathematics-II (Calculus, Ordinary Differential Equations and Complex Variable), Khanna Book Publishing Co.
2. Reena Garg, Engineering Mathematics, Khanna Book Publishing Company, 2022.
3. Reena Garg, Advanced Engineering Mathematics, Khanna Book Publishing Company, 2021.
4. Erwin Kreyszig, Advanced Engineering Mathematics, 10th Edition, John Wiley & Sons, 2006.

5. Veerarajan T., Engineering Mathematics for first year, Tata McGraw-Hill, New Delhi, 2008.
6. W. E. Boyce and R. C. DiPrima, Elementary Differential Equations and Boundary Value Problems, 9th Edition, Wiley India, 2009.
7. D. Poole, Linear Algebra: A Modern Introduction, 2nd Edition, Brooks/Cole, 2005.
8. S. L. Ross, Differential Equations, 3rd Edition, Wiley India, 1984.
9. E. A. Coddington, An Introduction to Ordinary Differential Equations, Prentice Hall India, 1995.
10. E. L. Ince, Ordinary Differential Equations, Dover Publications, 1958.
11. J. W. Brown and R. V. Churchill, Complex Variables and Applications, 7th Ed., Mc-Graw Hill, 2004.
12. N.P. Bali and Manish Goyal, A text book of Engineering Mathematics, Laxmi Publications, Reprint, 2008.
13. B.S. Grewal, Higher Engineering Mathematics, Khanna Publishers, 36th Edition, 2010.

Note: The modules have been prepared keeping the following from the Textbooks/References in mind:

(1) Module 1: Sections 7.3-7.5, 7.7, 7.8, 8.1-8.4 of [1].
(2) Module 2: Sections 1.4, 1.5 of [1]; Section 5.1 of [2].
(3) Module 3: Sections 2.5, 2.6, 2.10, 5.1, 5.3, 5.4, 5.5 of [1].
(4) Module 4: Sections 13.3 – 13.7, 17.1 – 17.3 of [1].
(5) Module 5: Sections 14.1 – 14.4, 15.2 – 15.4, 16.1 – 16.4 of [1].

COURSE OUTCOMES: The objective of this course is to familiarize the prospective engineers with techniques in matrices, ordinary differential equations and complex variables. It aims to equip the students to deal with advanced level of mathematics and applications that would be essential for their disciplines.

The students will learn:

- The essential tool of matrices and linear algebra in a comprehensive manner.
- The effective mathematical tools for the solutions of differential equations that model physical processes.

The tools of differentiation and integration of functions of a complex variable that are used in various techniques dealing engineering problems.

****

ESC-201	Programming for Problem Solving	3L:0T:4P	5 Credits

Course Objectives:

1. To learn the fundamentals of computers.
2. To understand the various steps in program development.
3. To learn the syntax and semantics of C programming language.
4. To learn the usage of structured programming approach in solving problems.
5. To understated and formulate algorithm for programming script
6. To analyze the output based on the given input variables

Course Contents:

Module I: Introduction to Programming**;** Introduction to components of a computer system (disks, memory, processor, where a program is stored and executed, operating system, compilers etc.)

Idea of Algorithm: steps to solve logical and numerical problems. Representation of Algorithm: Flowchart/Pseudocode with examples.

From algorithms to programs; source code, variables (with data types) variables and memory locations, Syntax and Logical Errors in compilation, object and executable code.

Module II: Arithmetic expressions and precedence.

Module III: Conditional Branching and Loops. Writing and evaluation of conditionals and consequent branching. Iteration and loops.

Module IV: Arrays, Arrays (1-D, 2-D), Character arrays and Strings

Module V: Basic Algorithms, Searching, Basic Sorting Algorithms (Bubble, Insertion and Selection), Finding roots of equations, notion of order of complexity through example programs (no formal definition required)

Module VI: Function, Functions (including using built in libraries), Parameter passing in functions, call by value, Passing arrays to functions: idea of call by reference

Module VII: Recursion, Recursion as a different way of solving problems. Example programs, such as Finding Factorial, Fibonacci series, Ackerman function etc. Quick sort or Merge sort.

Module VIII: Structures, Defining structures and Array of Structures

Module IX: Pointers, Idea of pointers, Defining pointers, Use of Pointers in self-referential structures, notion of linked list (no implementation)

Module X: File handling (only if time is available, otherwise should be done as part of the lab).

PRACTICALS:

1. Familiarization with programming environment
2. Simple computational problems using arithmetic expressions
3. Problems involving if-then-else structures
4. Iterative problems e.g., sum of series
5. 1D Array manipulation
6. Matrix problems, String operations
7. Simple functions
8. Programming for solving Numerical methods problems
9. Recursive functions
10. Pointers and structures
11. File operations

TEXT/REFERENCE BOOKS:

1. AICTE's Prescribed Textbook: Programming for Problem Solving, Khanna Book Publishing Co.
2. Byron Gottfried, Schaum's Outline of Programming with C, McGraw-Hill.
3. E. Balaguruswamy, Programming in ANSI C, Tata McGraw-Hill.
4. Brian W. Kernighan and Dennis M. Ritchie, The C Programming Language, Prentice Hall of India.

Alternative NPTEL/SWAYAM Course:

S. No.	NPTEL Course Name	Instructor	Host Institute
1	INTRODUCTION TO PROGRAMMING IN C	PROF. SATYADEV NANDAKUMAR	IIT KANPUR
2	PROBLEM SOLVING THROUGH PROGRAMMING IN C	PROF. ANUPAM BASU	IIT KHARAGPUR

EXPERIMENTS THAT MAY BE PERFORMED THROUGH VIRTUAL LABS:

S. No.	Experiment Name	Experiment Link(s)
1	Simple computational problems using arithmetic expressions.	http://ps-iiith.vlabs.ac.in/exp7/Introduction.html?do

		main=Computer%20Science&lab=Problem%20Solving%20Lab
2	Iterative problems e.g., sum of series.	http://ps-iiith.vlabs.ac.in/exp4/Introduction.html?domain=Computer%20Science&lab=Problem%20Solving%20Lab
3	1D Array manipulation.	http://cse02-iiith.vlabs.ac.in/exp4/index.html
4	Matrix problems, String operations.	http://ps-iiith.vlabs.ac.in/exp5/Introduction.html?domain=Computer%20Science&lab=Problem%20Solving%20Lab
5	Simple functions.	http://cse02-iiith.vlabs.ac.in/exp2/index.html
6	Programming for solving Numerical methods problems.	http://ps-iiith.vlabs.ac.in/exp1/Introduction.html?domain=Computer%20Science&lab=Problem%20Solving%20Lab
7	Recursive functions.	http://ps-iiith.vlabs.ac.in/exp6/Introduction.html?domain=Computer%20Science&lab=Problem%20Solving%20Lab

COURSE OUTCOMES: The student will learn following through lectures:

- To formulate simple algorithms for arithmetic and logical problems.
- To translate the algorithms to programs (in C language).
- To test and execute the programs and correct syntax and logical errors.
- To implement conditional branching, iteration and recursion.
- To decompose a problem into functions and synthesize a complete program using divide and conquer approach.
- To use arrays, pointers and structures to formulate algorithms and programs.
- To apply programming to solve matrix addition and multiplication problems and searching and sorting problems.
- To apply programming to solve simple numerical method problems, namely rot finding of function, differentiation of function and simple integration.

The student will learn following through Practicals:

- To formulate the algorithms for simple problems.
- To translate given algorithms to a working and correct program.
- To be able to correct syntax errors as reported by the compilers.
- To be able to identify and correct logical errors encountered at run time.
- To be able to write iterative as well as recursive programs.
- To be able to represent data in arrays, strings and structures and manipulate them through a program.
- To be able to declare pointers of different types and use them in defining self-referential structures.
- To be able to create, read and write to and from simple text files.

******

HSMC-201	English	2L:0T:2P	3 Credits

Course Objective:

- To provide learning environment to practice listening, speaking, reading and writing skills.
- To assist the students to carry on the tasks and activities through guided instructions and materials.
- To effectively integrate English language learning with employability skills and training.
- To provide hands-on experience through case-studies, mini-projects, group and individual presentations.

Course Content:

Module I: Vocabulary Building

1.1. The concept of Word Formation
1.2. Root words from foreign languages and their use in English
1.3. Acquaintance with prefixes and suffixes from foreign languages in English to form derivatives.
1.4. Synonyms, antonyms, and standard abbreviations.

Module II: Basic Writing Skills

1.1. Sentence Structures
1.2. Use of phrases and clauses in sentences
1.3. Importance of proper punctuation
1.4. Creating coherence
1.5. Organizing principles of paragraphs in documents
1.6. Techniques for writing precisely

Module III: Identifying Common Errors in Writing

1.1. Subject-verb agreement
1.2. Noun-pronoun agreement
1.3. Misplaced modifiers
1.4. Articles
1.5. Prepositions
1.6. Redundancies
1.7. Clichés

Module IV: Nature and Style of sensible Writing

1.1. Describing
1.2. Defining
1.3. Classifying
1.4. Providing examples or evidence
1.5. Writing introduction and conclusion

Module V: Writing Practices

1.1. Comprehension
1.2. Précis Writing
1.3. Essay Writing

Module VI: Oral Communication

(This Module involves interactive practice sessions in Language Lab)

- Listening Comprehension
- Pronunciation, Intonation, Stress and Rhythm
- Common Everyday Situations: Conversations and Dialogues
- Communication at Workplace
- Interviews
- Formal Presentations

Text/Reference Books:

1. **AICTE's Prescribed Textbook: English (with Lab Manual), Khanna Book Publishing Co.**
2. Effective Communication Skills. Kul Bhushan Kumar, Khanna Book Publishing, 2022.
3. Practical English Usage. Michael Swan. OUP. 1995.
4. Remedial English Grammar. F.T. Wood. Macmillan.2007
5. On Writing Well. William Zinsser. Harper Resource Book. 2001
6. Study Writing. Liz Hamp-Lyons and Ben Heasly. Cambridge University Press. 2006.
7. Communication Skills. Sanjay Kumar and PushpLata. Oxford University Press. 2011.
8. Exercises in Spoken English. Parts. I-III. CIEFL, Hyderabad. Oxford University Press.

Alternative NPTEL/SWAYAM Course:

S. No.	NPTEL Course Name	Instructor	Host Institute
1	ENGLISH LANGUAGE FOR COMPETITIVE EXAMS	PROF. AYSHA IQBAL	IIT MADRAS
2.	TECHNICAL ENGLISH FOR ENGINEERS	PROF. AYSHA IQBAL	IIT MADRAS

Course Outcomes: The student will acquire basic proficiency in English including reading and listening comprehension, writing and speaking skills.

*****

ESC-202	Workshop/Manufacturing Practices	1L:0T:4P	3 Credits

Course Objective:

1. To provide exposure to the students with hands on experience on various basic engineering practices in Civil, Mechanical, Electrical and Electronics Engineering.
2. To have a study and hands-on-exercise on plumbing and carpentry components.
3. To have a practice on gas welding, foundry operations and fitting
4. To have a study on measurement of electrical quantities, energy and resistance to earth.
5. To have a practice on soldering.

Course Content:

Module I: Manufacturing Methods- casting, forming, machining, joining, advanced manufacturing methods.

Module II: CNC machining, Additive manufacturing.

Module III: Fitting operations & power tools.

Module IV: Electrical & Electronics.

Module V: Carpentry.

Module VI: Plastic moulding, glass cutting.

Module VII: Metal casting.

Module VIII: Welding (arc welding & gas welding), brazing.

Practicals:

1. Machine shop
2. Fitting shop
3. Carpentry
4. Electrical & Electronics
5. Welding shop (Arc welding + Gas welding)
6. Casting
7. Smithy
8. Plastic moulding & Glass Cutting

Examinations could involve the actual fabrication of simple components, utilizing one or more of the techniques covered above.

Suggested Text/Reference Books:

1. AICTE's Prescribed Textbook: Workshop / Manufacturing Practices (with Lab Manual), Khanna Book Publishing Co.
2. Hajra Choudhury S.K., Hajra Choudhury A.K. and Nirjhar Roy S.K., "Elements of Workshop Technology", Vol. I 2008 and Vol. II 2010, Media promoters and publishers private limited, Mumbai**.**
3. Kalpakjian S. And Steven S. Schmid, "Manufacturing Engineering and Technology", 4th edition, Pearson Education India Edition, 2002.
4. Gowri P. Hariharan and A. Suresh Babu," Manufacturing Technology – I" Pearson Education, 2008.
5. Roy A. Lindberg, "Processes and Materials of Manufacture", 4th edition, Prentice Hall India, 1998.
6. Rao P.N., "Manufacturing Technology", Vol. I and Vol. II, Tata McGraw Hill House, 2017.

EXPERIMENTS THAT MAY BE PERFORMED THROUGH VIRTUAL LABS:

S. No.	Experiment Name	Experiment Link(s)
1	Welding shop (Arc welding + Gas welding).	http://mm-coep.vlabs.ac.in/LaserSpotWelding/Theory.html?domain=Mechanical%20Engineering&lab=Welcome%20to%20Micromachining%20laboratory
2	Casting	http://fab-coep.vlabs.ac.in/exp7/Theory.html?domain=Mechanical%20Engineering&lab=Welcome%20to%20FAB%20laboratory

Course Outcomes: Upon completion of this course, the students will gain knowledge of the different manufacturing processes which are commonly employed in the industry, to fabricate components using different materials.

Laboratory Outcomes:

Upon completion of this laboratory course, students will be able:

- To fabricate components with their own hands.
- To relate practical knowledge of the dimensional accuracies and dimensional tolerances possible with different manufacturing processes.
- To design small devices of their interest by assembling different components.

*****

AU-102	Sports and Yoga	2L:0T:0P	0 Credit

Course Objective(s):

- To make the students understand the importance of sound health and fitness principles as they relate to better health.
- To expose the students to a variety of physical and yogic activities aimed at stimulating their continued inquiry about Yoga, physical education, health and fitness.
- To create a safe, progressive, methodical and efficient activity based plan to enhance improvement and minimize risk of injury.
- To develop among students an appreciation of physical activity as a lifetime pursuit and a means to better health.

Course Contents:

Module I: Introduction to Physical Education

- Meaning & definition of Physical Education
- Aims & Objectives of Physical Education
- Changing trends in Physical Education

Module II: Olympic Movement

- Ancient & Modern Olympics (Summer & Winter)
- Olympic Symbols, Ideals, Objectives & Values
- Awards and Honours in the field of Sports in India (Dronacharya Award, Arjuna Award, Dhayanchand Award, Rajiv Gandhi Khel Ratna Award etc.)

Module III: Physical Fitness, Wellness & Lifestyle

- Meaning & Importance of Physical Fitness & Wellness
- Components of Physical fitness
- Components of Health related fitness
- Components of wellness
- Preventing Health Threats through Lifestyle Change
- Concept of Positive Lifestyle

Module IV: Fundamentals of Anatomy & Physiology in Physical Education, Sports and Yoga

- Define Anatomy, Physiology & Its Importance

- Effect of exercise on the functioning of Various Body Systems. (Circulatory System, Respiratory System, Neuro-Muscular System etc.)

Module V: Kinesiology, Biomechanics & Sports

- Meaning & Importance of Kinesiology & Biomechanics in Physical Edu. & Sports
- Newton's Law of Motion & its application in sports.
- Friction and its effects in Sports.

Module VI: Postures

- Meaning and Concept of Postures.
- Causes of Bad Posture.
- Advantages & disadvantages of weight training.
- Concept & advantages of Correct Posture.
- Common Postural Deformities – Knock Knee; Flat Foot; Round Shoulders; Lordosis, Kyphosis, Bow Legs and Scoliosis.
- Corrective Measures for Postural Deformities

Module VII: Yoga

- Meaning & Importance of Yoga
- Elements of Yoga
- Introduction - Asanas, Pranayama, Meditation & Yogic Kriyas
- Yoga for concentration & related Asanas (Sukhasana; Tadasana; Padmasana & Shashankasana)
- Relaxation Techniques for improving concentration - Yog-nidra

Module VIII: Yoga & Lifestyle

- Asanas as preventive measures.
- Hypertension: Tadasana, Vajrasana, Pavan Muktasana, Ardha Chakrasana, Bhujangasana, Sharasana.
- Obesity: Procedure, Benefits & contraindications for Vajrasana, Hastasana, Trikonasana, Ardh Matsyendrasana.
- Back Pain: Tadasana, Ardh Matsyendrasana, Vakrasana, Shalabhasana, Bhujangasana.
- Diabetes: Procedure, Benefits & contraindications for Bhujangasana, Paschimottasana, Pavan Muktasana, Ardh Matsyendrasana.
- Asthema: Procedure, Benefits & contraindications for Sukhasana, Chakrasana, Gomukhasana, Parvatasana, Bhujangasana, Paschimottasana, Matsyasana.

Module IX: Training and Planning in Sports

- Meaning of Training
- Warming up and limbering down
- Skill, Technique & Style
- Meaning and Objectives of Planning.
- Tournament – Knock-Out, League/Round Robin & Combination.

Module X: Psychology & Sports

- Definition & Importance of Psychology in Physical Edu. & Sports
- Define & Differentiate Between Growth & Development
- Adolescent Problems & Their Management
- Emotion: Concept, Type & Controlling of emotions
- Meaning, Concept & Types of Aggressions in Sports.
- Psychological benefits of exercise.
- Anxiety & Fear and its effects on Sports Performance.
- Motivation, its type & techniques.
- Understanding Stress & Coping Strategies.

Module XI: Doping

- Meaning and Concept of Doping
- Prohibited Substances & Methods
- Side Effects of Prohibited Substances

Module XII: Sports Medicine

- First Aid – Definition, Aims & Objectives.
- Sports injuries: Classification, Causes & Prevention.
- Management of Injuries: Soft Tissue Injuries and Bone & Joint Injuries

Module XIII: Sports / Games

Following subtopics related to any one Game/Sport of choice of student out of:

Athletics, Badminton, Basketball, Chess, Cricket, Kabaddi, Lawn Tennis, Swimming, Table Tennis, Volleyball, Yoga etc.

- History of the Game/Sport.
- Latest General Rules of the Game/Sport.
- Specifications of Play Fields and Related Sports Equipment.
- Important Tournaments and Venues.

- Sports Personalities.
- Proper Sports Gear and its Importance.

Text Books/References:

1. Modern Trends and Physical Education by Prof. Ajmer Singh.
2. Light On Yoga by B.K.S. Iyengar.
3. Health and Physical Education – NCERT (11th and 12th Classes)

Course Outcomes: On successful completion of the course the students will be able:

1. To practice Physical activities and Hatha Yoga focusing on yoga for strength, flexibility, and relaxation.
2. To learn techniques for increasing concentration and decreasing anxiety which leads to stronger academic performance.
3. To learn breathing exercises and healthy fitness activities
4. To understand basic skills associated with yoga and physical activities including strength and flexibility, balance and coordination.
5. To perform yoga movements in various combination and forms.
6. To assess current personal fitness levels.
7. To identify opportModuleies for participation in yoga and sports activities.
8. To develop understanding of health-related fitness components: cardiorespiratory endurance, flexibility and body composition etc.
9. To improve personal fitness through participation in sports and yogic activities.
10. To develop understanding of psychological problems associated with the age and lifestyle.
11. To demonstrate an understanding of sound nutritional practices as related to health and physical performance.
12. To assess yoga activities in terms of fitness value.
13. To identify and apply injury prevention principles related to yoga and physical fitness activities.
1. To understand and correctly apply biomechanical and physiological principles elated to exercise and training.

****

HSMC (H-102)	Universal Human Values-II: Understanding Harmony And Ethical Human Conduct	2L:1T:0P	3 Credits

Pre-requisites: None. Universal Human Values 1 (Desirable)

1-COURSES ON HUMAN VALUES

During the Induction Program, students would get an initial exposure to human values through Universal Human Values-I. This exposure is to be augmented by this compulsory full semester foundation course.

Objectives of UHV-II Course

This introductory course input is intended:

1. To help the students appreciate the essential complementarily between 'VALUES' and 'SKILLS' to ensure sustained happiness and prosperity which are the core aspirations of all human beings.
2. To facilitate the development of a Holistic perspective among students towards life and profession as well as towards happiness and prosperity based on a correct understanding of the Human reality and the rest of existence. Such a holistic perspective forms the basis of Universal Human Values and movement towards value-based living in a natural way.
3. To highlight plausible implications of such a Holistic understanding in terms of ethical human conduct, trustful and mutually fulfilling human behavior and mutually enriching interaction with Nature.

Thus, this course is intended to provide a much-needed orientational input in value education to the young enquiring minds.

Salient Features of the Course

The salient features of this course are:

1. It presents a universal approach to value education by developing the right understanding of reality (i.e. a worldview of the reality “as it is”) through the process of self-exploration.
2. The whole course is presented in the form of a dialogue whereby a set of proposals about various aspects of the reality are presented and the students are encouraged to self-explore the proposals by verifying them on the basis of their natural acceptance within oneself and validate experientially in living.
3. The prime focus throughout the course is toward affecting a qualitative transformation in the life of the student rather than just a transfer of information.
4. While introducing the holistic worldview and its implications, a critical appraisal of the prevailing notions is also made to enable the students discern the difference on their own right.

Course Methodology

1. The methodology of this course is explorational and thus universally adaptable. It involves a systematic and rational study of the human being vis-à-vis the rest of existence.
2. The course is in the form of 28 lectures (discussions) and 14 practice sessions.
3. It is free from any dogma or value prescriptions.
4. It is a process of self-investigation and self-exploration, and not of giving sermons. Whatever is found as truth or reality is stated as a proposal and the students are facilitated to verify it in their own right, based on their Natural Acceptance and subsequent Experiential Validation – the whole existence is the lab and every activity is a source of reflection.
5. This process of self-exploration takes the form of a dialogue between the teacher and the students to begin with, and then to continue within the student in every activity, leading to continuous self-evolution.
6. This self-exploration also enables them to critically evaluate their pre-conditionings and present beliefs.

2-COURSE TOPICS

The course has 28 lectures and 14 tutorials in 5 modules. The lectures and tutorials are of 01-hour duration. Tutorial sessions are to be used to explore and practice what has been proposed during the lecture sessions.

The Teacher's Manual provides the outline for lectures as well as practice sessions. The teacher is expected to present the issues to be discussed as propositions and encourage the students to have a dialogue.

The syllabus for the lectures and practice sessions is given below:

Module 1 – Introduction to Value Education (6 lectures and 3 tutorials for practice session)

Lecture 1: Right Understanding, Relationship and Physical Facility (Holistic Development and the Role of Education)

Lecture 2: Understanding Value Education

Tutorial 1: **Practice Session PS1** Sharing about Oneself

Lecture 3: Self-exploration as the Process for Value Education

Lecture 4: Continuous Happiness and Prosperity – the Basic Human Aspirations

Tutorial 2: Practice Session PS2 Exploring Human Consciousness

Lecture 5: Happiness and Prosperity – Current Scenario

Lecture 6: Method to Fulfill the Basic Human Aspirations

Tutorial 3: Practice Session PS3 Exploring Natural Acceptance

Expected outcome:

The students start exploring themselves: get comfortable with each other and with the teacher; they start appreciating the need and relevance for the course.

The students start finding that technical education without study of human values can generate more problems than solutions. They also start feeling that lack of understanding of human values is the root cause of most of the present-day problems; and a sustained solution could emerge only through understanding of value-based living. Any solution brought out through fear, temptation of dogma will not be sustainable.

The students are able to see that verification on the basic of natural acceptance and experiential validation through living is the only way to verify right or wrong, and referring to any external source like text or instrument or any other person cannot enable them to verify with authenticity; it will only develop assumptions.

The students are able to see that their practice in living is not in harmony with their natural acceptance most of the time, and all they need to do is to refer to their natural acceptance to overcome this disharmony.

The students are able to see that lack of right understanding leading to lack of relationship is the major cause of problems in their family and not the lack of physical facility in most of the cases, while they have given higher priority to earning of physical facility in their life giving less value to or even ignoring relationships and not being aware that right understanding is the most important requirement for any human being.

Module 2 – Harmony in the Human Being (6 lectures and 3 tutorials for practice session)

Lecture 7: Understanding Human being as the Co-existence of the Self and the Body

Lecture 8: Distinguishing between the Needs of the Self and the Body

Tutorial 4: Practice Session PS4 Exploring the difference of Needs of Self and Body

Lecture 9: The Body as an Instrument of the Self

Lecture 10: Understanding Harmony in the Self

Tutorial 5: Practice Session PS5 Exploring Sources of Imagination in the Self

Lecture 11: Harmony of the Self with the Body

Lecture 12: Programme to ensure self-regulation and Health

Tutorial 6: Practice Session PS6 Exploring Harmony of Self with the Body

Expected outcome:

The students are able to see that they can enlist their desires and the desires are not vague. Also they are able to relate their desires to 'I' and 'Body' distinctly. If any desire appears related to both, they are able to see that the feeling is related to I while the physical facility is related to the body. They are also able to see that 'I' and Body are two realities, and most of their desires are related to 'I' and not body, while their efforts are mostly centered on the fulfilment of the needs of the body assuming that it will meet the needs of 'I' too.

The students are able to see that all physical facility they are required for a limited time in a limited quantity. Also, they are able to see that in case of feelings, they want continuity of the naturally acceptable feelings and they do not want feelings which are not naturally acceptable even for a single moment.

The students are able to see that activities like understanding, desire, though and selection are the activities of 'I' only the activities like breathing, palpitation of different parts of the body are fully the activities of the body with the acceptance of 'I' while the activities they do with their sense organs like hearing through ears, seeing through eyes, sensing through touch, tasting through tongue and smelling through nose or the activities they do with their work organs like hands, legs etc. are such activities that require the participation of both 'I' and body.

The students become aware of their activities of 'I' and start finding their focus of attention at different moments. Also they are able to see that most of their desires are coming from outside (through preconditioning or sensation) and are not based on their natural acceptance

The students are able to list down activities related to proper upkeep of the body and practice them in their daily routine. They are also able to appreciate the plants wildly growing in and around the campus which can be beneficial in curing different diseases.

Module 3 – Harmony in the Family and Society (6 lectures and 3 tutorials for practice session)

Lecture 13: Harmony in the Family – the Basic Unit of Human Interaction

Lecture 14: 'Trust' – the Foundational Value in Relationship

Tutorial 7: Practice Session PS7 Exploring the Feeling of Trust

Lecture 15: 'Respect' – as the Right Evaluation

Tutorial 8: Practice Session PS8 Exploring the Feeling of Respect

Lecture 16: Other Feelings, Justice in Human-to-Human Relationship

Lecture 17: Understanding Harmony in the Society

Lecture 18: Vision for the Universal Human Order

Tutorial 9: Practice Session PS9 Exploring Systems to fulfil Human Goal

Expected outcome:

The students are able to note that the natural acceptance (intention) is always for living in harmony, only competence is lacking! We generally evaluate ourselves on the basis of our intention and others on the basis of their competence! We seldom look at our competence and others' intention as a result we conclude that I am a good person and other is a bad person.

The students are able to see that respect is right evaluation, and only right evaluation leads to fulfilment in relationship. Many present problems in the society are an outcome of differentiation (lack of understanding of respect), like gender biasness, generation gap, caste conflicts, class struggle, dominations through power play, communal violence, clash of isms and so on so forth. All these problems can be solved by realizing that the other is like me as he has the same natural acceptance, potential and program to ensure a happy and prosperous life for them and for others through he may have different body, physical facility or beliefs.

The students are able to use their creativity for education children. The students are able to see that they can play a role in providing value education for children. They are able to put in simple words the issues that are essential to understand for children and comprehensible to them. The students are able to develop an outline of holistic model for social science and compare it with the existing model.

Module 4 – Harmony in the Nature/Existence (4 lectures and 2 tutorials for practice session)

Lecture 19: Understanding Harmony in the Nature

Lecture 20: Interconnectedness, self-regulation and Mutual Fulfilment among the Four Orders of Nature

Tutorial 10: Practice Session PS10 Exploring the Four Orders of Nature

Lecture 21: Realizing Existence as Co-existence at All Levels

Lecture 22: The Holistic Perception of Harmony in Existence

Tutorial 11: Practice Session PS11 Exploring Co-existence in Existence

Expected outcome:

The students are able to differentiate between the characteristics and activities of different orders and study the mutual fulfilment among them. They are also able to see that human being s are not fulfilling to other orders today and need to take appropriate steps to ensure right participation (in terms of nurturing, protection and right utilization) in the nature.

The students feel confident that they can understand the whole existence; nothing is a mystery in this existence. They are also able to see the interconnectedness in the nature, and point out how different

courses of study relate to the different units and levels. Also, they are able to make out how these courses can be made appropriate and holistic.

Module 5 – Implications of the Holistic Understanding – a Look at Professional Ethics (6 lectures and 3 tutorials for practice session)

Lecture 23: Natural Acceptance of Human Values

Lecture 24: Definitiveness of (Ethical) Human Conduct

Tutorial 12: Practice Session PS12 Exploring Ethical Human Conduct

Lecture 25: A Basis for Humanistic Education, Humanistic Constitution and Universal Human Order

Lecture 26: Competence in Professional Ethics

Tutorial 13: Practice Session PS13 Exploring Humanistic Models in Education

Lecture 27: Holistic Technologies, Production Systems and Management Models-Typical Case Studies

Lecture 28: Strategies for Transition towards Value-based Life and Profession

Tutorial 14: Practice Session PS14 Exploring Steps of Transition towards Universal Human Order

Expected outcome:

The students are able to present sustainable solutions to the problems in society and nature. They are also able to see that these solutions are practicable and draw roadmaps to achieve them.

The students are able to grasp the right utilization of their knowledge in their streams of Technology/Engineering/Management/any other area of study to ensure mutual fulfilment. E.g. mutually enriching production system with rest of nature.

The students are able to sincerely evaluate the course and share with their friends. They are also able to suggest measures to make the course more effective and relevant. They are also able to make use of their understanding in the course for the happy and prosperous family and society.

Guidelines and Content for Practice Sessions (Tutorials)

In order to connect the content of the proposals with practice (living), 14 practice sessions have been designed. The full set of practice sessions is available in the Teacher's Manual as well as the website.

Practice Sessions for Module 1 – Introduction to Value Education

PS1 Sharing about Oneself

PS2 Exploring Human Consciousness

PS3 Exploring Natural Acceptance

Practice Sessions for Module 2 – Harmony in the Human Being

PS4 Exploring the difference of Needs of Self and Body

PS5 Exploring Sources of Imagination in the Self

PS6 Exploring Harmony of Self with the Body

Practice Sessions for Module 3 – Harmony in the Family and Society

PS7 Exploring the Feeling of Trust

PS8 Exploring the Feeling of Respect

PS9 Exploring Systems to fulfil Human Goal

Practice Sessions for Module 4 – Harmony in the Nature (Existence)

PS10 Exploring the Four Orders of Nature

PS11 Exploring Co-existence in Existence

Practice Sessions for Module 5 – Implications of the Holistic Understanding – a Look at Professional Ethics

PS12 Exploring Ethical Human Conduct

PS13 Exploring Humanistic Models in Education

PS14 Exploring Steps of Transition towards Universal Human Order

As an example, PS7 is a practice session in module 3 regarding trust. It is explained below:

PS7: Form small groups in the class and in that group initiate dialogue and ask the eight questions related to trust. The eight questions are:

1a. Do I want to make myself happy?

1b. Am I able to make myself always happy?

2a. Do I want to make the other happy?

2b. Am I able to make the other always happy?

3a. Does the other want to make him happy?	3b. Is the other able to make him always happy?
4a. Does the other want to make me happy?	4b. Is the other able to make me always happy?
Intention (Natural Acceptance)	Competence
What is the answer?	What is the answer?

Let each student answer the questions for himself/herself and everyone else. Discuss the difference between intention and competence. Observe whether you evaluate your intention and competence as well as the others' intention and competence.

Expected outcome of PS7: The students are able to see that the first four questions are related to our Natural Acceptance i.e. intention and the next four to our Competence. They are able to note that the intention is always correct, only competence is lacking! We generally evaluate ourselves on the basis of our intention and others on the basis of their competence! We seldom look at our competence and others' intention, as a result we conclude that I am a good person and other is a bad person.

3-READINGS:

3-1-Text Book and Teachers Manual

a. The Textbook - A Foundation Course in Human Values and Professional Ethics, R R Gaur, R Asthana, G P Bagaria, 2nd Revised Edition, Excel Books, New Delhi, 2019. ISBN 978-93-87034-47-1

b. The Teacher's Manual- Teachers' Manual for A Foundation Course in Human Values and Professional Ethics, RR Gaur, R Asthana, G P Bagaria, 2nd Revised Edition, Excel Books, New Delhi, 2019. ISBN 978-93-87034-53

c. Professional Ethics and Human Values, Premvir Kapoor, ISBN: 978-93-86173-652, Khanna Book Publishing Company, New Delhi, 2022.

3-2-Reference Books

1. JeevanVidya: EkParichaya, A Nagaraj, JeevanVidyaPrakashan, Amarkantak, 1999.
2. Human Values, A.N. Tripathi, New Age Intl. Publishers, New Delhi, 2004.
3. The Story of Stuff (Book).
4. The Story of My Experiments with Truth - by Mohandas Karamchand Gandhi
5. Small is Beautiful - E. F Schumacher.
6. Slow is Beautiful - Cecile Andrews
7. Economy of Permanence - J C Kumarappa
8. Bharat Mein Angreji Raj – Pandit Sunderlal

9. Rediscovering India - by Dharampal
10. Hind Swaraj or Indian Home Rule - by Mohandas K. Gandhi
11. India Wins Freedom - Maulana Abdul Kalam Azad
12. Vivekananda - Romain Rolland (English)
13. Gandhi - Romain Rolland (English)

4-MODE OF CONDUCT (L-T-P-C 2-1-0-3)

Lecture hours are to be used for interactive discussion, placing the proposals about the topics at hand and motivating students to reflect, explore and verify them.

Tutorial hours are to be used for practice sessions.

While analysing and discussing the topic, the faculty mentor's role is in pointing to essential elements to help in sorting them out from the surface elements. In other words, help the students explore the important or critical elements.

In the discussions, particularly during practice sessions (tutorials), the mentor encourages the student to connect with one's own self and do self-observation, self-reflection and self-exploration.

Scenarios may be used to initiate discussion. The student is encouraged to take up" ordinary" situations rather than" extra-ordinary" situations. Such observations and their analyses are shared and discussed with other students and faculty mentor, in a group sitting.

Tutorials (experiments or practical) are important for the course. The difference is that the laboratory is everyday life, and practical are how you behave and work in real life. Depending on the nature of topics, worksheets, home assignment and/or activity are included. The practice sessions (tutorials) would also provide support to a student in performing actions commensurate to his/her beliefs. It is intended that this would lead to development of commitment, namely behaving and working based on basic human values.

It is recommended that this content be placed before the student as it is, in the form of a basic foundation course, without including anything else or excluding any part of this content. Additional content may be offered in separate, higher courses.

This course is to be taught by faculty from every teaching department.

Teacher preparation with a minimum exposure to at least one 8-day Faculty Development Program on Universal Human Values is deemed essential.

5-SUGGESTED ASSESSMENT:

This is a compulsory credit course. The assessment is to provide a fair state of development of the student, so participation in classroom discussions, self-assessment, peer assessment etc. will be used in evaluation.

Example:

Assessment by faculty mentor: 10 marks

Self-assessment: 10 marks

Assessment by peers: 10 marks

Socially relevant project/Group Activities/Assignments: 20 marks

Semester End Examination: 50 marks

The overall pass percentage is 40%. In case the student fails, he/she must repeat the course.

6-OUTCOME OF THE COURSE:

By the end of the course, students are expected to become more aware of themselves, and their surroundings (family, society, nature); they would become more responsible in life, and in handling problems with sustainable solutions, while keeping human relationships and human nature in mind.

They would have better critical ability. They would also become sensitive to their commitment towards what they have understood (human values, human relationship and human society). It is hoped that they would be able to apply what they have learnt to their own self in different day-to-day settings in real life, at least a beginning would be made in this direction.

Therefore, the course and further follow up is expected to positively impact common graduate attributes like:

1. Holistic vision of life
2. Socially responsible behaviour
3. Environmentally responsible work
4. Ethical human conduct
5. Having Competence and Capabilities for Maintaining Health and Hygiene
6. Appreciation and aspiration for excellence (merit) and gratitude for all

This is only an introductory foundational input. It would be desirable to follow it up by

a) Faculty-student or mentor-mentee programs throughout their time with the institution

b) Higher level courses on human values in every aspect of living.

****

SEMESTER – III

SEMESTER III

PCC CS-301	Data Structure and Algorithms	3L:0T:4P	5 Credits

Learning outcomes of the course (i.e. statements on students' understanding and skills at the end of the course the student shall have):

Essential:

1. Understanding abstract specification of data-structures and their implementation.
2. Understanding time and space complexity of programs and data-structures.
3. Knowledge of basic data-structures, their applications and relative merits.
4. Ability to convert an algorithmic solution to a program using suitable data-structures and analyse the trade-offs involved in terms of time and space complexity.

Desirable/Advanced:

1. Amortized complexity
2. Use of randomization in data-structures

Detailed contents for Essential Learning Outcomes:

Module	Topics	Pedagogy teaching suggestions	Nature of lab / assignment / practice
Module 1: Introduction and basic terminology (1 week)	(i) Notion of data-structures and algorithms. (ii) $logn, n, 2^n$: understanding growth of these functions, and applications (binary search and extensions to similar problems) (iii) Worst-case, average-case time/space complexity and their relative merits. (iv) Asymptotic Notation: $O(\ ), \Omega(\ )$	T1: Chapter 3 (i) Explain the interplay between algorithms and data-structures (ii) Explain the meaning of worst and average case (iii) Examples of $O(\ )$, Motivation behind asymptotic analysis (large n and ignoring constants). (iv) Discuss recurrence relation for binary search.	(i) Worst/average case analysis for small pseudo-codes (ii) Prove/disprove why a function $f(n)$ is $O(g(n))$ (and similarly for Ω). (iii) Variations on binary search with applications, recursive and iterative implementation of binary search with applications to problems.

Module 2: Abstract Data-types, Arrays, Linked Lists, Stacks, Queues Dictionary ADT, Trees, Binary Trees (2.5 weeks)	(i)Abstract data-type (ADTs): arrays and linked list ADTs. (ii) Stacks, Queues: ADTs and implementations using arrays, linked lists. (iii) Doubly linked lists: ADT and implementation (iv) Dictionary ADT: implementation using array, linked lists, binary search. (v) Tree ADT and examples (vi) Implementation of trees and basic traversal algorithms (vii) Binary trees and in order traversal	T1: chapter 4, Chapter 6 (i) Explain the difference between specification and implementation of ADTs. (ii) Discuss stack implementation using arrays of fixed size and linked list, and relative merits. (iii) Circular array and linked list implementation of queues, and relative merits. (iv) Discuss some applications of stacks: post-fix notation, matching parentheses etc. (v) Dictionary ADT: discuss why neither arrays nor linked list is a good implementation, but sorted arrays are good for searching. (vi) Discuss situations where we need to store hierarchical data. (vii) Discuss pre-order and post-order, traversals and applications in finding height and similar problems.	(i) Implementation of stacks with application to a problem. (ii) Implementation of queues with application to a problem. (iii) Implementation of trees with applications for storing and accessing hierarchical data.
Module 3: Priority Queues and Heaps	(i) Priority Queue ADT (ii) Definition of heaps	T1: Chapter 7.1-7.3 (i) Start with FindMin and Insert, and a simple	(i) Array implementation of

(1 week)	(iii) Implementation of Priority Queues using heaps and running time analysis (iv) Implementation of heaps using arrays. (v) Heap-sort	$O(1)$ time and $O(1)$ space algorithm. Buildup towards DeleteMin. (ii) Discuss algorithm for heaps: inserting an element, modifying an element, and deleting the minimum element. (iii) Discuss why array implementation is more efficient than balanced binary trees	heaps and application to a problem. (ii) k-ary heaps: compare with binary heaps (both in theory and practice)
Module 4: Binary Search Trees, AVL Trees, 2-4 trees (3 weeks)	(i) Binary Search Trees: definition and some basic algorithms. (ii) Implementation of Dictionary ADTs using Binary Search trees and running time analysis (iii) AVL trees: height balance condition, rotations, and implementation of dictionary ADT (iv) 2-4 Trees: Multi-way search trees, implementation of dictionary ADT, Informal discussion of extension to B-trees.	T1: Chapter 9.1, 9.2, 9.4 (i) Discuss algorithms for finding predecessor or successor, and similar problems in Binary Search Trees. (ii) Explain why height of a Binary Search Tree may not remain $O(logn)$. (iii) Explain how the height balance condition ensures that the height is $O(logn)$, and how rotation changes the structure of a tree. (iv) Explain why rotations in AVL trees restore height balance condition (v) Explain why 2-4 trees have $O(logn)$height and the running time of insert/delete operations. (vi) Discuss how balanced binary tree data-structures can	(i) Implementation of AVL trees with search, insert, delete operations and application to a problem. Comparison with unbalanced Binary Search Trees. (ii) Implementation of 2-4 trees with search, insert, delete operations and application to a problem. (iii) Comparison of the two implementations above.

		implement a priority queue	
Module 5: Hash tables, tries (2 weeks)	(i) Map ADT (ii) Hash Tables and implementation of Map using Hash Tables (iii) Design of hash functions (iv) Collision resolution schemes: chaining, open addressing schemes like linear probing, quadratic probing, double hashing. (v) Applications of Hashing: finding duplicates, set intersection, etc. (vi) Tries: implementation of Map ADT using tries. (vii) Compressed tries and suffix tries.	T1: Chapter 8.1-8.3, Chapter 11.3 (i) Explain the difference between Map and Dictionary ADT. (ii) Discuss how hash functions can have non-numeric keys as input. (iii) Discuss the relative merits of hash tables and balanced binary search trees. (iv) Discuss how hashing can be a substitute for sorting in many cases. (v) Explain why tries can be better than balanced binary search trees in some settings. (vi) Explain how compressed tries save space (vii) Discuss real-life applications of tries.	(i) Implementation of hash tables with applications to a problem. (ii) Implementation of tries and applications to a problem.
Module 6: Sorting, Selection (1.5 weeks)	(i) Bubble sort, insertion sort, selection sort. (ii) Merge sort and divide and conquer paradigm (iii) Quick sort: average and worst case analysis, randomized quicksort (intuitive explanation)	T1: Chapter 10.1, 10.2, 10.4, 10.5, 10.7 (i) Discuss why $O(n^2)$ time algorithms can be useful sometimes (small data, data nearly sorted etc.) (ii) Only the recurrence for merge sort and mention that divide-and-conquer paradigm will	Implementation of sorting algorithms and comparison of running times on large data-sets.

	(iv) Selection based on partitioning ideas used in Quick Sort.	be explored more in algorithms course. (iii) Discuss the randomized splitting algorithm for quicksort and selection and explain intuitively the expected running time.	
Module 7: Graphs, representations and traversal algorithms, applications of BFS, DFS (2.5 weeks)	(i) Graph ADTs and applications (ii) Adjacency list and adjacency matrix representations and relative merits (iii) Basic graph definitions: paths, cycles, trees, spanning trees, connected components, Euler's formula. (iv) Depth First Search Traversal algorithm for directed graphs: classification of edges into forward, back and cross edges. Applications to cycle finding, topological sort in directed acyclic graphs, finding connected components. Running time analysis. (v) Breadth first search algorithm: implementation using queues, shortest path tree property. Running time analysis	T1: Chapter 12.1-12.4 (i) Discuss the wide applicability of graphs including social networks, internet. (ii) Discuss time and space complexity of basic operations using adjacency list and adjacency matrix. (iii) Discuss why trees have $n-1$edges. (iv) Discuss how DFS can be thought of as exploration with backtracking. Explain the role of stack in DFS. (v) Explain how BFS can be thought of as traversal along shortest paths and implementation using queues. (vi) Formal proof of why BFS yields a shortest path tree.	(i) Graph implementation using adjacency list and DFS/BFS traversal with applications.

Detailed Contents for Desirable Learning Outcomes (optional, <= 3 modules):

Module	Topics	Pedagogy teaching suggestions	Nature of lab / assignment / practice
Module 1: Amortized Complexity (1 week)	(i) Binary counter (ii) Binomial Heaps (iii) Extendible arrays	T1: Chapter 5.1.3, R2: Chapter 17.1 (i) Explain the motivation behind amortized analysis. (ii) Analyze amortized complexity by explicit calculation of total number of operations after $n$steps.	(i) Exercises on amortized time complexity (e.g., a queue using two stacks etc.) (ii) Implementation of extendible arrays.
Module 2: Randomization in data-structures (1 week)	(i) Skip Lists (ii) Randomized quick-sort	T1: Chapter 8.4, 10.2 (i)Explain the idea of expectation of a random variable. (ii) How does expected running time translate to actual running time (iii)Recurrence for expected running time randomized quicksort	(i) Exercises on calculation of expectation of a random variable. (ii) Implementation of skip lists and comparison with AVL trees.

Suggested text books / Online lectures or tutorials:

1. "Expert Data Structures with C" by R.B. Patel, Khanna Book Publishing Company, New Delhi, 4th Edition.

2. "Data Structures and Algorithms in Java", by Michael T. Goodrich and Roberto Tamassia, John Wiley & Sons; 3rd Edition.

3. "Data Structures and Algorithms in Python", by Michael T. Goodrich and Robert, Tamassia, Wiley, 1st Edition.

4. "Taming Python by Programming", Jeeva Jose, Khanna Book Publishing Company.

5. "Fundamentals of Data Structures", Sartaj Saini, University Press.

6. In case any other programming language is used for this course, some other suitable text book may be chosen.

Suggested reference books / Online resources:

1. NPTEL video series, Data-structures and Algorithms, Instructor: Naveen Garg.
2. Introduction to Algorithms, 4th Edition, Thomas H Cormen, Charles E Lieserson, Ronald L Rivest and Clifford Stein, MIT Press/McGraw-Hill.

The Discipline Graduate Attributes (GAs) to which this course contributes significantly: *CS1, CS2, CS4*

*****

SEMESTER – IV

SEMESTER IV

PCC CS-401	Discrete Mathematics	3L:1T:0P	4 Credits

Prerequisites remark:
Familiarity with some mathematical notation, ideas and concepts covered at the pre-college levels.

Learning Outcomes of the course (i.e. statements on students' understanding and skills at the end of the course the student shall have):

Essential:

1. Understand examples in Computer Science through mathematical terminology and notation.
2. Construct direct, and indirect, proofs of basic theorems.
3. Understand the differences between a mathematical proof, a heuristic, and a conjecture.
4. Learn how to divide a problem, or a proof, into smaller cases.
5. Formulate mathematical claims and be able to construct counterexamples.
6. Apply the knowledge of mathematics to solve real-world problems.

Desirable/Advanced:

1. Identify formal algebraic structures in computer science.
2. Work with probability & statistics in a rigorous way.
3. Use these course topics to design, and rigorously analyze, real-life algorithms.

Detailed contents for Essential Learning Outcomes:

Module	Topics	Pedagogy / teaching suggestions	Nature of lab / assignment / practice
Module 1: Set, Relations, Functions. (~1.5 weeks)	Operations and Laws of Sets, Cartesian Products, Binary Relation, Partial Ordering Relation, Equivalence Relation, Image of a Set, Sum and Product of Functions, Bijective functions, Inverse and Composite Function, Size of a Set, Finite and infinite Sets, Countable and uncountable Sets, Cantor's diagonal argument and The Power Set theorem.	- Chapter 1,3 of T1. - Chapter 2 of T2 - Discuss some examples from computer science: +Uncountability of Reals +Uncomputability by the diagonal argument (ref. Chap.6 of T1).	Make assignments using the books. To test: (1) what was done in the class (2) whether the student can think and apply the concepts.

		+Relational database (ref. Chap.3 of T1. Refer R8).	
Module 2: Proof strategies. (1.5 weeks)	Proof Methods and Strategies: Forward Proof, Proof by Contradiction, Proof by Contraposition, Proof of Necessity and Sufficiency, Case analysis, Induction.	- Chapter 1 of T1. - Chapter 1 of T2. - Discuss some examples from computer science: +Examples like above. +√2 is irrational. +Prime numbers are infinite (Chap.4.3 of T2).	Make assignments using the books. To test: (1) what was done in the class (2) whether the student can think and apply the concepts.
Module 3: Modular Arithmetic. (1.5 weeks)	Extended Euclid's Greatest Common Divisor algorithm, The Fundamental Theorem of Arithmetic, Modular arithmetic, Coprimality (or Euler's totient function), Chinese Remainder Theorem.	- Chapter 4 of T2. Chapter 1.20 of T1. - References T5 & R5. - Discuss some examples from computer science: +Diffie-Hellman key-exchange. +RSA Cryptosystem.	Make assignments using the books. To test: (1) what was done in the class (2) whether the student can think and apply the concepts.
Module 4: Combinatorics. (1.5 weeks)	Permutation & Combination, Inclusion-Exclusion, Pigeon-hole principle, Generating functions, Recurrence.	- Chapter 2, 8, 9 of T1. Chapter 6 of T2. Chapter 2, 3, 4, 5, 6 of T4. Also refer to R3, R4. - Discuss some examples from computer science: +Count binary trees. +Count matched parentheses.	Make assignments using the books. To test: (1) what was done in the class (2) whether the student can think and apply the concepts.

		+Count matrix chain multiplication.	
Module 5: Graphs. (1.5 weeks)	Connected components, Paths, Cycles, Trees, Hamiltonian/ Eulerian Walks, Coloring, Planarity, Matching.	- Chapter 4, 5 of T1. Chapter 10, 11 of T2. - Discuss some examples from computer science: +Shortest-distance (Chapter 7 of T1). +Minimum spanning tree. +Prefix codes. +Graph isomorphism problem.	Make assignments using the books. To test: (1) what was done in the class (2) whether the student can think and apply the concepts.
Module 6: Logic. (1.5 weeks)	Languages of Propositional logic and First-order logic, expressing natural language sentences in languages of propositional and first-order logic, expressing natural language predicates in the language of first-order logic. Semantics of First-order logic: interpretation and its use in evaluating a formula. Optional advanced topics if there is extra time: Semantic entailment, Validity and Satisfiability. What is a proof system? E.g. natural deduction or analytical tableau. Notions of Consistency and Completeness of a logic.	- Chapter 1, 11 of T1. Chapter 1, 12 of T2. Also refer to R2. - Discuss some examples from computer science (also, refer T3): +SAT solvers and why these are so useful. +Relational Calculus. +Language for specification like Z. +Hoare logic for program verification.	Make assignments using the books. To test: (1) what was done in the class (2) whether the student can think and apply the concepts.

Detailed Contents for Desirable Learning Outcomes (optional, <= 3 modules):

Module	Topics	Pedagogy teaching suggestions	Nature of lab / assignment / practice
Module 7: Algebra. (1.5 weeks)	Group, Permutation Groups, Cosets, Normal Subgroups, Ring, Field, Finite fields, Fermat's little theorem, Homomorphisms, Isomorphisms.	- Chapter 10 of T1. Also, refer R5, R6. - Discuss some examples from computer science (ref. T3): +RSA Cryptosystem. +Reed-Solomon error-correction encoding.	Make assignments using the books. To test: (1) what was done in the class (2) whether the student can think and apply the concepts.
Module 8: Discrete probability. (1.5 weeks)	Topics to be taught from the viewpoint of CS instead of Maths: Discrete Sample Space, Probability Distribution, Random variables, Expectation, Variance, Bernoulli trials, Conditional probability & independence (Bayes' Theorem).	- Chapter 2 of T1. Chapter 7 of T2. - Discuss some examples from computer science (also refer R7): +Randomized algorithms. +Heuristics.	Make assignments using the books. To test: (1) what was done in the class (2) whether the student can think and apply the concepts.

Suggested text books / Online lectures or tutorials:

T1. Singh, S.B., Discrete Mathematics, Khanna Book Publishing Company, New Delhi.

T2. Liu, C. L., & Mohapatra, D. P. (2008). Elements of Discrete Mathematics. Tata McGraw-Hill.

T3. Rosen, K. H. (2019). Discrete Mathematics and Its Applications. (8th Edition) ISBN10: 125967651X ISBN13: 9781259676512.

T4. Huth, M., & Ryan, M. (2004). Logic in Computer Science: Modelling and Reasoning about Systems (2nd ed.). Cambridge University Press.

T5. Cohen, D. I. A. (1978). Basic techniques of combinatorial theory. John Wiley.

T6. Niven, I., Zuckerman, H. S., & Montgomery, H. L. (1991). An introduction to the theory of numbers. John Wiley & Sons.

T7. Sarkar, Discrete Mathematics and Its Applications, Oxford Press.

Suggested reference books / Online resources:

R1. Norman L. Biggs, Discrete Mathematics, (2nd ed. 2002), Oxford University Press.

R2. Smullyan, R. M. (1995). First-order logic. Courier Corporation.

R3. Bóna, M. (2006). A walk through combinatorics: an introduction to enumeration and graph theory.

R4. Cameron, P. J. (1994). Combinatorics: topics, techniques, algorithms. Cambridge University Press.

R5. Shoup, V. (2009). A computational introduction to number theory and algebra. Cambridge University Press.

R6. Herstein, I. N. (2006). Topics in algebra. John Wiley & Sons.

R7. Mitzenmacher, M., & Upfal, E. (2017). Probability and computing: Randomization and probabilistic techniques in algorithms and data analysis. Cambridge University Press.

R8. C. J. Date (2019). Database Design and Relational Theory. Normal Forms and All That Jazz.

Prepared by:

1. Somenath Biswas, Computer Science & Engineering Department, IIT Goa (Retd. IIT Kanpur)
2. Partha Mukhopadhyay, Theoretical Computer Science Group, Chennai Mathematical Institute
3. Nitin Saxena, Computer Science & Engineering Department, IIT Kanpur
4. Bhabani P. Sinha, Advanced Computing & Microelectronics Unit, Indian Statistical Institute Kolkata

The Discipline Graduate Attributes (GAs) to which this course contributes significantly: CS4

***Other discipline GAs to which this course may contribute somewhat*:** CS2, CS5, CS6

*****

PCC CS-402	Computer Organization & Architecture	3L:0T:4P	5 Credits

Learning Outcomes of the course (i.e. statements on students' understanding and skills at the end of the course the student shall have):

Essential:

1. The key components of a basic computer.
2. The key components of a CPU and how the instructions are executed.
3. Execution and time taken by instructions in a pipelined processor.
4. The need for memory hierarchy and efficiency achieved due to the use of cache.
5. How the data is stored and input-output is performed in computers.

Desirable/Advanced:

1. Super-scalar and multi-core architectures for crossing one clock per instruction barrier
2. Cache data coherence related challenges in multi-core processors

Detailed contents for Essential Learning Outcomes:

Module (appx duration in weeks)	**Topics**	**Pedagogy / teaching suggestions**	**Nature of lab / assignment / practice**
Module 1: Introduction (Lectures 6) (Weeks 2)	Role of abstraction, basic functional units of a computer, Von-Neumann model of computation, A note on Moore's law, Notion of IPC, and performance. Data representation and basic operations.	T1: Chapter 1 R1: Chapter 1(Section 1.7.2) R3: Chapter 1 R2: Chapter 3	
Module 2: Instruction Set Architecture (RISC-V) (Lectures 8/9) (Weeks 3)	CPU registers, instruction format and encoding, addressing modes, instruction set, instruction types, instruction decoding and execution, basic instruction cycle, Reduced Instruction Set Computer (RISC), Complex Instruction Set Computer	R1: Chapter 3, 5 R3: Chapter 2, Chapter 5 (Section 5.1 to 5.5) R7: Chapter 3	Lab Modules 1 and 4

	(CISC), RISC-V instructions; X86 Instruction set.		
Module 3: The Processor (Lectures 6) (Weeks 2)	Revisiting clocking methodology, Amdahl's law, Building a data path and control, single cycle processor, multi-cycle processor, instruction pipelining, Notion of ILP, data and control hazards and their mitigations.	T1: Chapter 1, 4 R3: Chapter 6	Lab Modules 2 and 4
Module 4: Memory hierarchy (Lectures 8/9) (Weeks 3)	SRAM/DRAM, locality of reference, Caching: different indexing mechanisms, Trade-offs related to block size, associativity, and cache size, Processor-cache interactions for a read/write request, basic optimizations like write-through/write-back caches, Average memory access time, Cache replacement policies (LRU), Memory interleaving.	T1: Chapter 5 R3: Chapter 8	Lab Module 2
Module 5: Storage and I/O (Lectures 6) (Weeks 2)	Introduction to magnetic disks (notion of tracks, sectors), flash memory. I/O mapped, and memory mapped I/O. I/O data transfer techniques: programmed I/O, Interrupt-driven I/O, and DMA.	R5: Chapter 5 (Section 5.4) Chapter 6 (Section 6.1) Chapter 7 (Section 7.1 -7.5)	Lab Modules 3 and 5

Detailed Contents for Desirable Learning Outcomes (optional, <= 3 modules):

Module	Topics	Pedagogy teaching suggestions	Nature of lab / assignment / practice
Module 6: Superscalar processors and multicore systems (Lectures 6) (Weeks 2)	Limits of ILP, SMT processors, Introduction to multicore systems and cache coherence issues	Lab Modules 1 and 4	

Laboratory Modules

The laboratory component consists of 5 modules out of which three can be chosen based on the overall curriculum and its emphasis. Modules 4 and 5 assume that students have a background in using FPGA kits in their Digital Electronics course (ESC302) and have also been introduced in programming in one of the HDLs (VHD or Verilog).

Possible three options are:

1. Module 1 + Module 2 + Module 3
   Instructions & assembly language + basic performance + advanced performance analysis
2. Architecture + Module 1 + Module 2 + Module 4
   Instructions & assembly language + basic performance + basic processor design
   Module 1 + Module 4 + Module 5
3. Instructions & assembly language + basic processor design + I/O and architecture enhancements.

Detailed contents for Essential Learning Outcomes:

Module (appx duration in weeks)	Topics	Comments
Module 1: (Weeks 4) Objective: Understanding architecture and instructions through assembly programming	Write programs in ARM/RISC V assembly language and test these on an instruction set simulator. Typical examples are given below. Some of these are dependent on I/O facilities provided by the simulator. • Generate some interesting numbers (example - Happy numbers, Autonomic numbers, Hardy-Ramanujan numbers etc.) • Implement a 4-function calculator • Sort an integer array using merge sort (recursive)	Essential component

	• Evaluate an arithmetic expression specified as a string (using recursive functions) • Implement a simple game	
Module 2: Understanding performance issues related to pipelining and cache using architectural simulator (Weeks 4/5)	Usage of an instruction pipeline visualization tool like RIPES Write or generate sequence of instructions and observe the overall pipeline stalls with and without data hazards, control hazards, and with/without data forwarding. Rearrange the sequence of instructions or the program so that the pipeline stalls will be minimized.	Optional
Module 3: Understanding memory access patterns and changing basic cache memory parameters to analyze the impact of standard programs or benchmarks using architectural simulators. (Weeks 3)	Configure the simulator [gem5 is preferred] to operate on the binaries of the benchmark as the input. Run the program and examine the IPC, cache hit rate, number of conflict misses and block replacements. Vary the cache size, block size, and associativity and analyze the metrics and reason the changes observed. Modify the block replacement algorithms and see the impact at cache memory performance Calculate the access time, power and are associated with a given cache configuration. Vary the cache size, block size, and associativity and analyze the metrics and reason the changes observed.	Optional: Familiarity with tools like GEM5, CACTI and PIN, and access to benchmarks like SPEC, PARSEC, SPLASH. Any other tools/simulators that can support memory pattern analysis is also fine.
Module 4: (Weeks 4/5) Objective: Understanding computer architecture by designing a CPU on FPGA kit	Design a simple ARM/RISC V processor for a small subset of instructions and implement on FPGA board. This is done in stages as follows. • Design CPU for the instruction subset {add, sub, cmp, mov, ldr, str, beq, bne, b}, with each instruction executing in a single cycle. No sequential control	Optional Prerequisite: Use of FPGA kits as well as exposure to VHDL/Verilog

	required. For each instruction, only a limited set of variants are considered. ● Use memory generator to add program and data memories. ● Include circuit for single step execution and for displaying signals of interest. ● Modify the design to allow multi-cycle execution of instructions. ● Enhance the design to include all DP instructions and all variants of the second operand.	
Module 5: Objective: Understanding computer architecture performance issues as well as I/O through architectural enhancements (on FPGA) (Weeks 4/5)	Extension of the CPU design and I/O programming ● Enhance the design to include all variants of DT instructions. ● Implement multiply group of instructions. ● Enhance the design to implement "branch and link" instruction and include full predication. ● Include limited exception handling. ● Interface 7-segment display and 4x4 keypad. ● Demonstrate execution of simple programs.	Optional Prerequisite: Use of FPGA kits as well as exposure to VHDL/Verilog

Suggested text books / Online lectures or tutorials:

T1 "Computer Organization and Design: The Hardware/Software Interface", David A. Patterson and John L. Hennessy, 5th Edition, Elsevier.

Suggested reference books / Online resources:

R1 "Computer Organisation & Architecture", Smruti Ranjan Sarangi, McGraw Hill

R2 "Computer System Architecture", Mano M. Morris, Pearson.

R3 "Computer Organization and Embedded Systems", 6th Edition by Carl Hamacher, McGraHill Higher Education

R4 "Computer Architecture and Organization", 3rd Edition by John P. Hayes, WCB/McGraw-Hill

R5 "Computer Organization and Architecture: Designing for Performance", 10th Edition by William Stallings, Pearson Education.

R6 "Computer System Design and Architecture", 2nd Edition by Vincent P. Heuring and Harry F. Jordan, Pearson Education.

R7 http://web.cecs.pdx.edu/~harry/riscv/RISCV-Summary.pdf

Online simulators and tools:

RIPES: https://freesoft.dev/program/108505982

GEM5: https://www.gem5.org/documentation/learning_gem5/introduction/

CACTI: https://github.com/HewlettPackard/cacti

PIN: https://www.intel.com/content/www/us/en/developer/articles/tool/pin-a-binary-instrumentation-tool-downloads.html

TEJAS: https://www.cse.iitd.ac.in/~srsarangi/archbooksoft.html

XILINX (VHDL/Verilog tools): https://www.xilinx.com/support/university/students.html

Prepared by:

1. Prof. Anshul Kumar, IIT Delhi
2. Prof. Anupam Basu, IIT Kharagpur/NIT Durgapur
3. Prof. Indranil Sengupa, IIT Kharagpur
4. Prof. Biswabandan Panda, IIT Bombay
5. Prof. John Jose, IIT Guwahati
6. Prof. M. Balakrishnan, IIT Delhi

The Discipline Graduate Attributes (GAs) to which this course contributes significantly: CS3 (and also to some graduate attributes for ECE)

***Other discipline GAs to which this course may contribute somewhat*:** CS4, CS6

******

PCC CS-404	Design and Analysis of Algorithms	3L:0T:4P	5 Credits

Learning Outcomes of the course (i.e. statements on students' understanding and skills at the end of the course the student shall have):

Essential:

1. Analyse the asymptotic performance of algorithms.
2. Establish the correctness of algorithms.
3. Demonstrate familiarity with major algorithms and data structures.
4. Apply important algorithmic design paradigms and methods of analysis.
5. Synthesize efficient algorithms in common engineering design situations.
6. Understanding limits of efficient computation.

Desirable/Advanced:

1. Intuitive understanding of what makes a problem NP-hard.
2. Apply network flow techniques to algorithm design.

Detailed contents for Essential Learning Outcomes:

Module (appx duration in weeks)	**Topics**	**Pedagogy / teaching suggestions**	**Nature of lab / assignment / practice**
Module 1: Applications of Graph Search (~1.5 weeks)	(i) Review of BFS/DFS (ii) Checking if an undirected graph is 2-edge connected. (iii) Checking if a directed graph is strongly connected.	T1: Chapter 3, T2: Chapter 3, R1: Chapter 22. Review breadth first search (BFS) and depth first search (DFS) in undirected and directed graphs; Introduce notion of a bridge and cut-vertex.	Checking if a graph is bi connected
Module 2: Greedy algorithms (2 weeks)	(i) Introduction to the greedy paradigm (ii) Examples of activity selection, deadline scheduling, fractional knapsack, Kruskal's algorithm for minimum spanning trees, Huffman coding.	T1: Chapter 4, T2: Chapter 5. Exchange arguments are a useful recipe for proving correctness of greedy algorithms. Illustrate these through examples. Show examples where	(i) Prim's algorithm for minimum spanning trees. (ii) Examples where greedy algorithms are not optimal.

		greedy does not give an optimum solution	
Module 3: Divide and Conquer (2 weeks)	(i) Explain why the divide and conquer paradigm is useful. (ii) Illustrate the paradigm through integer multiplication. (iii) Writing recurrence relations and solving them. (iv) Further examples from geometry – domination number of a set of points, identifying maximal points, closest pair of points. (v) Linear time algorithm for finding the median. (vi) Randomized divide and conquer algorithms: randomized quicksort and selection.	T1: Chapter 5, T2: Chapter 2. Solve recurrences by building the recurrence tree. Motivate choices of parameters in median finding algorithm by showing how recurrence tree changes. Mention expectation of a random variable and how to make sense of running time of randomized algorithms.	(i) Solve some recurrence relations. (ii) Modify discussed algorithms (e.g., dividing into three parts instead of two parts, or two unequal parts, etc.) and analyse using recurrences. (iii) Some elementary exercises on expectation calculation.
Module 4: Dynamic Programming and shortest paths (2.5 weeks)	(i) Computing Fibonacci numbers and why divide-and-conquer is not a good idea. Idea of storing function calls, tables. (ii) Notion of sub problems and optimal substructure. (iii) Illustration through subset sum, (integer) knapsack, longest increasing subsequence, longest common subsequence, matrix chain multiplication. Dijkstra's algorithm for single-source shortest paths, Bellman-Ford for SSSP with negative weights, Floyd Warshall for APSP.	T1: Chapter 6, T2: Chapter 4, Chapter 6. Discuss why Dijkstra's algorithm is an example of dynamic programming. Extending Bellman-Ford to APSP and to find negative cycles. Discuss how dynamic programming problems can often be cast as longest paths in acyclic graphs.	Exercises on dynamic programming (textbook problems)

Module 5: Network flows (2 weeks)	The maximum s-t flow problem in capacitated networks. Ford Fulkerson algorithm or maximum flow. Max-flow min-cut theorem and integrality of maximum flow for integral capacities. Applications of max flow to maximum bipartite matching, max disjoint paths	T1: Chapter 7. 1-7.2, 7.5, 7.6 Notion of residual capacities and residual graphs and how this allows us to correct greedy decisions made in earlier steps of FF algorithm	(i) Some simple examples. (ii) Implementation of Ford Fulkerson algorithm.
Module 6: Intractability (2wks)	(i) Models of computation, Turing machines, RAM model. Brief discussion on other models of computation e.g. PRAM model, Memory Hierarchy etc. (ii) Notion of polynomial time computation. (iii) Polynomial time reductions. Yes and No instances of decision problems. Decision vs optimization. (iv) NP as a class of problems with Yes certificates which can be efficiently checked. (v) NP-hardness and Cook-Levin theorem (just the statement). (vi) NP-completeness. Examples of Reductions.	T1: Chapter 8, T2: Chapter 8 (i) Emphasize how reduction can be used to solve a problem using an algorithm for a different problem. (ii) Emphasize the asymmetry in the definition of NP: No efficiently checkable NO certificates for problems in NP. (iii) Problems which are NP-hard but not in NP. (iv) Examples of poly time reductions. Polytime as a measure of efficiency. Hardness only for the general instance, whereas special instances can be efficiently solvable.	Exercises on reductions, NP-completeness.

Suggested Texts:

1. Algorithm Design, 1st Edition, Jon Kleinberg and Éva Tardos, Pearson.
2. Algorithms, Sanjoy Dasgupta, Christos Papadimitriou, Umesh Vazirani
3. Design and Analysis of Algorithms, Gajendra Sharma, (4th Edition), Khanna Publishing House, New Delhi.

Suggested reference books:

1. Introduction to Algorithms, 4th Edition, Thomas H Cormen, Charles E Lieserson, Ronald L Rivest and Clifford Stein, MIT Press/McGraw-Hill.
2. Algorithm Design: Foundations, Analysis, and Internet Examples, Second Edition, Michael T Goodrich and Roberto Tamassia, Wiley.

Detailed Contents for Desirable Learning Outcomes (optional, <= 3 modules):

Module	Topics	Pedagogy teaching suggestions	Nature of lab / assignment / practice
Module 1:	Polynomial multiplication using FFT and DFT.	T1: Chapter 5.5, T2: Chapter 2.5 (i) Explain the interplay between the two representations of a polynomial. (ii) Explain why complex numbers arise naturally here.	Take two polynomials and multiply them by first computing the point-value representation using FFT and then recovering product using inverse FFT
Module 2:	Beyond NP-completeness: approximations, exponential algorithms, popular heuristics.	T1: Chapter 10, T2: Chapter 9 (i) Show that TSP can be solved in $2^n\ poly(n)$ time (rather than n! Time), and vertex cover in $2^k\ poly(n)$, where $k$is the size of the optimal vertex cover. (ii) Give examples of some simple approximation algorithms, e.g., bin packing, vertex cover. (iii) Popular heuristics for satisfiability.	(i) Implementation of some exponential time algorithms and heuristics and see how they scale with large n.

The Discipline Graduate Attributes (GAs) to which this course contributes significantly: *CS1, CS2, CS4*

******

PCC CS-405	Advanced Programming	3L:1T:0P	4 Credits

Familiarity with data structures, and introduction to programming.

Learning Outcomes of the course (i.e., statements on students' understanding and skills at the end of the course the student shall have):

Essential:

1. Understanding the build system: IDE, tools for testing, debugging, profiling, and source code management.
2. Students are able to demonstrate proficiency in object-oriented programming.
3. Identify and abstract the programming task involved for a given programming problem.
4. Learning and using language libraries for building large programs.
5. Ability to apply defensive programming techniques (e.g., assertions, exceptions).

Desirable/Advanced:

1. Ability to implement basic event-driven programming.
2. Understanding of the fundamentals of parallel programming.
3. Understanding of the basics of cloud computing.

Detailed contents for Essential Learning Outcomes:

Module (appx duration in weeks)	Topics	Pedagogy/teaching suggestions	Nature of lab / assignment / practice
Module 1: Familiarity with the programming environment (~1 weeks)	Understanding the build system, IDE, debugging, profiling (Eclipse TPTP / gprof / VTune etc.), and source code management	R5;	Familiarity with terminal/command prompt, using git commands and github to pull/commit/push/merge code, writing, compiling and running simple programs, debugging by setting breakpoints
Module 2: Basic principles of the object-oriented development process	Introduction to Object-Oriented Paradigm: Data encapsulation, modularity, code reuse, identifying classes, attributes, methods and objects, class relationships	Chapters 2-3, T1; Chapter 10, T2; Chapter 7, R;	Importing pre-written classes using the this keyword, calling and defining methods, writing and instantiating classes, setter/getter methods, instance variables, returning values, debugging using print function,

(~1.5 weeks)		Chapters 11-12, R1;	containment and association, scope and parameter passing
Module 3: Advanced features of OOP (~ 3 weeks)	Interfaces, inheritance, polymorphism, abstract classes, immutability, copying and cloning objects	Chapters 2-3, T1; Chapter 10, T2; Chapter 7, R4; Chapters 11-12, R1;	Parameter polymorphism, method resolution, declared v/s actual type, partially and fully overriding methods, calling superclass constructor from child class constructor, protected fields and methods, using an abstract parent class v/s an interface with default and abstract methods, object equality check, object comparison (Comparable/Comparator interface), Cloneable interface/copy constructor
Module 4: Unit testing (~0.5 weeks)	Unit testing, developing test suite	R6; R7;	JUnit/Boost.test testing framework, assertion methods, testcase timeout, testing for exceptions, test suite
Module 5: Using language APIs (~1 weeks)	Language supported libraries for handling advanced data structures	Section 13.2, T2; Section 13.7, R1;	Big-O notation, Java collection framework (or Boost libraries), sorting objects, iterating over objects
Module 6: Defensive programming (~1 weeks)	Exception handling, assertions	Section 9.4, T2; Chapter 14, R1; Section 2.7, R4;	Exception handling using try/catch block, nesting try/catch blocks, throw and throws keywords, rethrowing exceptions, handling checked exception, user defined exceptions

Module 7: Modeling and Design patterns (~2 weeks)	Basic modeling techniques – e.g. Class diagram, sequence diagram, use case diagrams, etc. Introduction to design patterns: iterator, singleton, flyweight, adapter, strategy, template, prototype, factory, façade, decorator, composite, proxy, chain of responsibility, observer, state)	Chapter 5, T1; Chapters 3-9, R3;	UML may be used for modeling.

Suggested text books:

T1. Taming Python by Programming, Jeeva Jose, Khanna Book Publishing Company, New Delhi.

T2. Grady Booch, Robert A. Maksimchuk, Michael W. Engle, Bobbi J. Young, Jim Conallen, Kelli A. Houston. Object-Oriented Analysis and Design with Applications.

T3. M. Scott. Programming Language Pragmatics. 4th edition.

Suggested reference books / Online resources:

- R1. Jeeva Jose, Introduction to Computing and Problem Solving with Python, Khanna Book Publishing Company, New Delhi.
- R2 R. Sebesta. Concepts of Programming Languages. 10th edition
- R3. J. Rumbaugh et al. The Unified Modeling Language Reference Manual.
- R4. Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, and Grady Booch. Design Patterns: Elements of Reusable Object-Oriented Software.
- R5. P. Van Roy and S. Haridi. Concepts, Techniques, and Models of Computer Programming.
- R6. https://missing.csail.mit.edu/
- R7. https://www.baeldung.com/junit
- R8. https://www.tutorialspoint.com/junit/index.htm
- R9. For UML tools, open source tools may be used (e.g. www.starUML.io, argouml.tigris.org/)

Detailed Contents for Desirable Learning Outcomes:

Option-1: Ability to implement basic event-driven programming-

Module	Topics	Pedagogy teaching suggestions	Nature of lab / assignment / practice
Module-1: Introduction to event-driven programming (~1 weeks)	Adding UI controls, GUI events, event handling	O1; Chapter 10, O2;	Create GUI forms/application with different GUI components (buttons, text boxes, etc.), responding to user input by creating event handlers, productivity in creating event handlers using anonymous classes and lambda methods
Module-2: Advanced GUI programming (~2 weeks)	Creating animations, connection with the database	O1;	Implement simple 2-D game that uses animations (timelines, moving shapes), client-server programming (multiplayer game), object serialization/deserialization (save/reload game)

Suggested text books / Online lectures or tutorials:

1. Oracle documentation: https://docs.oracle.com/javase/8/javase-clienttechnologies.htm
2. P. Van Roy and S. Haridi. Concepts, Techniques, and Models of Computer Programming.

Option-2: Understanding of the fundamentals of parallel programming

Module	Topics	Pedagogy teaching suggestions	Nature of lab / assignment / practice
Module-1: Introduction to multithreading (~1 weeks)	Ahmdal's law, speedup, parallel efficiency, thread creation	Chapters 2 and 4, T2;	Creating and joining threads, concurrency decomposition, assigning tasks to threads, calculate parallel speedup and

			efficiency, debugging multithreaded programs
Module-2: Thread pool (~1 weeks)	Task parallelism and thread pools (e.g., Java ForkJoin framework, OpenMP/CilkPlus/TBB in C++)	Section 13.2, T1; Section 13.7, R1;	Asynchronous task creation and joining, task parallelism for recursive parallelism, task dependency, controlling task granularity, comparing performance/productivity of explicit multithreading v/s using thread pools
Module-3: Mutual exclusion (~1 weeks)	Race conditions, deadlocks, producer-consumer problem	Chapter 13, T1; Chapter 4, T2; Chapter 13, R1; Chapter 8, R2;	Multithreaded program to push/pop items from shared queue - using monitor/Mutex locks, conditional wait and signaling, volatile keyword

Suggested text books / Online lectures or tutorials:

T1. M. Scott. Programming Language Pragmatics. 4th edition
T2. P. Pacheco. An Introduction to Parallel Programming.

Suggested reference books / Online resources:

R1. R. Sebesta. Concepts of Programming Languages. 10th edition
R2. P. Van Roy and S. Haridi. Concepts, Techniques, and Models of Computer Programming.

Option-3: Understanding of the basics of cloud computing

Module	**Topics**	**Pedagogy teaching suggestions**	**Nature of lab / assignment / practice**
Module 1:	Basics, Need, Advantages/Benefits, models	T1, Ch. 1, 2, 3	Set up some pre-installed programs and infrastructure to

Basic concepts plus a distributed programming framework such as MR/Spark (2 weeks)	such as PaaS/SaaS/IaaS/, distributed/cloud computing architectures Programming frameworks such as Map-Reduce or Spark, how to write programs in these programs, features such as fault-tolerance, using commodity hardware,	Illustrate the differences between M-R style programming and C-style programming	see how to run programs in the cloud. Programming in Map-Reduce/Spark to study scalability. Typical problems are from matrix multiplication, text processing, graphs , web crawling, and the like
Module 2: (1 week)	Key challenges in cloud/distributed computing: communication vs computation/ problem decomposition/ failures/ etc. and analyze the communication required of basic algorithms such as matrix multiplication	T2, Ch. 3. T3, Ch. 1	Redo exercises from week 2 to measure communication time

Suggested text books / Online lectures or tutorials:

T1: Data-Intensive Text Processing with MapReduce, Jimmy Lin and Chris Dyer, Morgan & Claypool Publishers, 2010.

T2: Parallel Computer Architecture, David Culler, J. P. Singh, and A. Gupta, Elsevier, 1998.

T3. Distributed Computing: Principles, Algorithms, and Systems, A.D. Kshemkalyani, M. Singhal, Cambridge University Press, March 201

Prepared by:

1. Vivek Kumar, Computer Science & Engineering Department, IIIT Delhi
2. Kishore Kothapalli, Computer Science & Engineering Department, IIIT Hyderabad
3. Swarnendu Biswas, Computer Science & Engineering Department, IIT Kanpur

The Discipline Graduate Attributes to which this course contributes significantly: *CS1, CS3, CS7, CS8*

1. ***Other discipline GAs to which this course may contribute somewhat:*** *CS2*

******

SEMESTER – V

SEMESTER V

PCC CS-505	Introduction to Database Systems	3L:0T:4P	5 Credits

Learning Outcomes of the course (i.e. statements on students' understanding and skills at the end of the course the student shall have):

Essential:

1. Ability to design and implement database schema for an application using RDBMS concepts.
2. Ability to write SQL queries for tasks of various complexities.
3. Ability to write an application program that uses a database system as the backend.
4. Understanding of internal working of a DBMS including data storage, indexing, query processing, transaction processing, concurrency control and recovery mechanisms.
5. Awareness of non-relational and parallel/distributed data management systems with a focus on scalability.

Desirable/Advanced:

[Nil]

Detailed contents for Essential Learning Outcomes:

Module (appx duration in weeks)	Topics	Pedagogy / teaching suggestions	Nature of lab / assignment / practice
Module 1: Introduction (~ 1 week)	(i) Motivation (ii) Introduction to Data Models (Relational, Semi structured, ER)		
Module 2: Relational Databases (~ 1. 5 weeks)	(i) Relational Data Model (ii) Relational Algebra (iii) Relational Calculus or Connection to First Order Logic (Optional)		(i) Simple pen + paper, and using Relax Relational Algebra calculator in browser
Module 3:	(i) DDL (ii) Insert/Delete/Update		(i) Laboratory exercises where students write SQL queries for various tasks.

SQL + interacting with database (~3 weeks)	(iii) Simple Queries (select/project/join/ aggregate queries) (iv) Complex queries (With Clause, Nested Subqueries, Views) (v) Programming in a standard language and interfacing with a DB backend		Platform can be PostgreSQL preferably, or MySQL. W3Schools/SQLite in web browser can also be used but beware of non-standard SQL features. (ii) Practice interfacing with a database from a program using connectors like JDBC/ODBC…
Module 4: Big Data (~1 week)	Key-value Stores and Semi-structured Data, using JSON and MongoDB, or other combinations		(i) Small exercises on MongoDB
Module 5: Database Design (~2 week)	(i) Introduction to ER model (ii) Mapping from ER to relational model (iii)Functional Dependencies (iv) Normalization (BCNF, Optionally 3NF)		(i) Exercise in ER design for an application starting with natural language description (ii) Convert ER design to tables (iii) Pen-and-paper exercises with FDs and normalization
Module 6: Physical Design (~2 week)	(i) Overview of Physical Storage (Hard Disks, Flash/SSD/RAM), sequential vs random I/O, Reliability via RAID (ii) Storage Organization (Records, Pages and Files), Database Buffers, Database Metadata (iii) Indexing, B+-Trees		(i) Use a B+-tree visualization system to understand how B+-trees work
Module 7: Query Processing and Optimization (~2 week)	(i) Query Processing: External sort, Joins using nested loops, indexed nested loops (ii) Overview of Query Optimization: equivalent expressions, and concept of cost-based optimization		(i) Examine query plans for sample queries by using the Explain feature of database systems. (ii) Small exercises to show benefit of indices.

Module 8: Transaction Processing (~2 week)	(i) Concept of transactions and schedules, ACID properties (ii) Conflict-serializability (iii) Concurrency control: locks, 2PL, Strict 2PL, optional: isolation levels (iv) Recovery using undo and redo logs		(i) Pen-and-paper exercises on conflicts, cycles, conflict serializability, recoverability, etc.

Suggested text books / Online lectures or tutorials:

1. Database System Concepts, 7th Edition, Silberschatz, Korth and Sudarshan, McGraw-Hill. Indian Edition released 2021
2. Fundamentals of Database Systems, 7th Edition, Elmasri and Navathe, Pearson Pubs, 2017
3. Principles of Database Management, Lemahieu, Broucke and Baesens, Cambridge University Press, 2018
4. Database Management Systems, RP Mahapatra, Khanna Publishing House, 2020.
5. Database Management Systems, Krishnan, McGraw Hill.

Suggested reference books / Online resources:

1. Software
   a. Relax Relational algebra calculator: https://dbis-uibk.github.io/relax/landing
   b. SQL: PostgreSQL/MySQL/MariaDB, or SQLite in browser
   c. B+-tree visualization: https://www.cs.usfca.edu/~galles/visualization/BPlusTree.html
   d. MongoDB
   e. Various DB systems playground: https://www.pdbmbook.com/playground

The Discipline Graduate Attributes (GAs) to which this course contributes significantly: *CS3, CS5, CS6*

Other discipline GAs to which this course may contribute somewhat*: CS1, CS7, CS9*

*****

PCC CS-603	Machine Learning	3L:1T:0P	4 Credits

Learning Outcomes of the course (i.e., statements on students' understanding and skills at the end of the course the student shall have):

Essential:

1. Understanding popular ML algorithms with their associated mathematical foundations.
2. Capability to implement basic algorithms using basic (python) libraries. Have hands-on experience in applying ML to problems encountered in various domains. Have exposure to high level ML libraries or frameworks such as TF, pytorch.
3. Make aware of the role of data in the future of computing and solving real-world problems.
4. Helping them connect/map real-world problems to the appropriate ML algorithm(s) to solve them
5. Appreciate the mathematical background behind the popular ML algorithms
6. Have awareness about the importance of core CS principles such as algorithmic thinking and systems design in ML

Desirable/Advanced:

1. Have a solid mathematical understanding of the popular ML algorithms
2. Preparedness to use state of the art machine learning algorithms in formulating and solving new problems.
3. Capability to train (or solve optimization problems) ML models with applications in real world use cases.

Detailed contents for Essential Learning Outcomes:

Module (appx duration of 3 weeks or 9-12 Hours)	Topics	Pedagogy / teaching suggestions	Nature of lab / assignment / practice
Introduction to ML (~2weeks)	(i) Motivation and role of machine learning in computer science and problem-solving (ii) Representation (features), linear transformations, Appreciate linear transformations and matrix vector operations in the	(i) Connect machine learning to the broader theme of Computer Science (ii) Expose broad canvas of machine learning; brief history and importance	(i)Experiments/notebooks/code that refresh Python, programming frameworks used for the course (ii) Experiments/Code that allows students to appreciate mathematics and data manipulation. Appreciate (a) Features, Representation of the

	context of data and representation. (iii) Problem formulations (classification and regression). (iv) Appreciate the probability distributions in the context of data, Prior probabilities and Bayes Rule. (v) Introduce paradigms of Learning (primarily supervised and unsupervised. Also a brief overview of others)	(iii) Role of data, Connection to the knowledge /experience in learning. (iv) Show successful examples of machine learning in Industry/working (v) Motivate students by Showing how ML and Data driven solutions could help in our day to day problems around (interdisciplinary such as agriculture, healthcare, education, living etc.) (vi) Refresh the basic mathematical notions that students may know (vectors, matrices, probabilities, etc.) with examples in ML (vii) Make students aware of relevant topics like "what is learnable"? And "what are the disadvantages of data driven solutions".	data/real-world phenomena (b) mathematical operations or transformations that manipulate the data (c) plot/visualise the data distributions (say in 2D) (d) Eigen values, eigen vectors, rank of matrices. (iii) Lab/Experiments that appreciate the problem of Classification and problem of Regression (iv) Lab/Experiments that appreciates the notions related to "Training" and "Testing" by considering algorithms like decision trees, nearest neighbour as black boxes.
Fundamentals of ML (~3 weeks)	(i) PCA and Dimensionality Reduction, (ii) Nearest Neighbours and KNN. (iii) Linear Regression (iv) Decision Tree Classifiers (v) Notion of Generalization and concern of Overfitting	(i) Focus on mathematical and algorithmic precise description of the content. (ii) Insights into these algorithms, why? When? What are the limitations? Why multiple algorithms exist for a specific problem (iii) Insights into the notion of generalization. Challenges for generalization. Assumptions to make	(i) Dimensionality Reduction using PCA and its applications in (a) removing irrelevant features (b) compression /compaction (c) efficient ML pipeline (ii) Experiment related to Nearest neighbour classifier, (a) visualize the decision boundaries (b) appreciate the role of hyper parameter K. Role of validation data in choice of hyper parameters

	(vi) Notion of Training, Validation and Testing; Connect to generalisation and overfitting.	(iv) Practical insights and tops on avoiding overfitting.	(iii) Decision Tree as a classifier and see the overfitting with "deep" trees. How the overfitting can be controlled by seeing validation performance during the training.
Selected Algorithms (~4 weeks)	(i) Ensembling and RF (ii) Linear SVM, (iii) K Means, (iv) Logistic Regression (v) Naive Bayes	(i) Make students appreciate the role of optimization in machine learning. Challenges in optimization and why we are sometimes happy with sub-optimal solutions. How assumptions make the algorithms simple/tractable. (ii) Make students appreciate the role of uncertainty in data and machine learning problems/solutions. Give probabilistic insights into Loss functions (em MSE, cross entropy) (iii) Introduce iterative algorithms, convergence, role of initialization etc. in a class of ML solutions. (iv) Connect the geometric view of Margin (e.g., linear SVM) and Probabilistic View of Margin (Logistic Regression) and the need of Generalization	(i) Experiments related to K-Means, by varying in "K", "initialization". How the "analysis of the algorithm" can be seen in the lab (e.g. change of objective across iterations). Try multiple datasets. Appreciate that "unsupervised discovery" makes sense in the problem under consideration. (ii) An experiment that demonstrates how SVM can yield a solution better than a simple linear separating solution. Appreciate the role of support vectors. Appreciate how SVMs extend to problems even if data is not linearly separable. (iii) An experiment that makes students appreciate the utility of naive Bayes classifier in practice (say designing a text classifier).

Neural Network Learning (~4 weeks)	(i) Role of Loss Functions and Optimization, (ii) Gradient Descent and Perceptron/Delta Learning, (iii) MLP, (iv)Backpropagation (v) MLP for Classification and Regression, (vi) Regularisation, Early Stopping (vii) Introduction to Deep Learning (viii) CNNs	(i) Appreciate (a) the neuron model (b) the neural network and its utility in modelling and solving the problem. Connect to the biological motivations and parallelism. (ii) Expose the simple elegant optimization scheme of gradient descent with associated mathematical rigour and insights. (iii) Expose the practical issues in extending GD to multiple layers and how the backpropagation algorithm efficiently computes the gradients. (iv) Expose the practical challenges in training a neural network (such as non-convexity, initialization, size of data, number of parameters) and how they are taken care of in the practical implementations of today. (v) Appreciate the need for empirical skills in training neural networks.	(i) Experiment that exposes the GD and BP in simple neural networks. Show the learning process (graphs) and performances. (ii) Experiment that use a modern library and implementation of a deep neural network, expose computational graphs, expose the generalized way of appreciating BP as a learning algorithm in Deep Neural Networks (iii) Experiment that uses a popular CNN architecture for practical application (say image classification). (iv) Experiments that strengthen the empirical skills in training with (a) initializations (b) update strategies (c) regularisation (d) multi fold validation on a small/medium size deep neural network that can be trained in 5 minutes.

Detailed Contents for Desirable Learning Outcomes (optional, <= 3 modules):

Module	Topics	Pedagogy teaching suggestions	Nature of lab / assignment / practice
Key Concepts from ML	Kernels (with SVM), Bayesian Methods, Generative Methods, HMM, EM, PAC learning	Focus on mathematical and analytical skills. Expose the intuition behind these algorithms. Introduce analysis of machine learning using a PAC model.	Python notebooks that demonstrate the use of these algorithms on public datasets
Deep Learning Architectures	Popular CNN Architectures, RNNs, GANS and Generative Models,	Introduce popular architectures, models, and the use of it in various settings.	(i) Use of popular architectures for pre-trained features and transfer learning (ii) Use of RNNs in learning "language models" in large text corpus (charRNN) (iii) Capability and practical challenged in working with GANS
Training Todays Neural Networks	Advances in Backpropagation and Optimization for Neural Networks Adversarial Learning	Appreciate the challenges in large nonconvex optimization and how many of today's design choices have helped.	(i) See how (a) initialization (b) momentum (c) update rules have helped in getting better minima/soln. (ii) Experience how regularisation helps in avoiding overfitting and getting better solutions

Notes on Exercises/Labs/Home works:

1. Use iPython/Jupyter notebooks to hand out assignments; such notebooks allow embedding instructions/videos/etc., making it easy for instructors, students as well as TAs
2. It may be good to have both theory and programming components in the assignment/homework component, to allow students to appreciate and learn both aspects of machine learning
3. Based on target audience, have a healthy mix of using an off-the-shelf machine learning library (e.g. using sklearn's decision tree function) and writing an algorithm from scratch (e.g. coding up decision tree from scratch)
4. Consider having a Kaggle-style hackathon in the course, where students get used to competitive machine learning.

Notes on Sequencing Lectures:

1. Consider your target audience (undergrad vs grad; 2nd-year vs 3rd/4th-year students; etc.) in deciding the prerequisites that the students may fulfil, and also the topics that you may want to cover.
2. It may be worthwhile visiting publicly available course outlines to verify and confirm that no important topics are missed out from both fundamental and contemporary perspectives (it is important to keep both these considerations in mind – to raise the spark of fundamental curiosity and also to provide utilitarian value).
3. Sequencing can be done considering the target audience in mind – one option may be to start with simple algorithms and gradually move towards mathematically involved ones.
4. Sequencing can also be modified based on whether there are any other machine learning or AI or related courses in the curriculum/elective list.

Suggested text books / Online lectures or tutorials:

1. Jeeva Jose, Introduction to Machine Learning, Khanna Book Publishing Company, New Delhi
2. Marc Peter Deisenroth, A. Aldo Faisal, Cheng Soon Ong, Mathematics for Machine Learning, Cambridge University Press (23 April 2020)
3. Tom M. Mitchell- Machine Learning - McGraw Hill Education, International Edition
4. Aurélien Géron Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, O'Reilly Media, Inc. 2nd Edition

Reference Books:

1. Ian Goodfellow, Yoshoua Bengio, and Aaron Courville Deep Learning MIT Press Ltd, Illustrated edition
2. Christopher M. Bishop Pattern Recognition and Machine Learning - Springer, 2nd edition

3. Trevor Hastie, Robert Tibshirani, and Jerome Friedman - The Elements of Statistical Learning: Data Mining, Inference, and Prediction - Springer, 2nd edition

4. Rajiv Chopra, Machine Learning, Khanna Book Publishing Company, New Delhi

Prepared by:

1. Mitesh Khapra, IIT Madras
2. Vineeth N Balasubramanian, IIT Hyderabad
3. P. K. Biswas, IIT Kharagpur
4. Piyush Rai, IIT Kanpur
5. Preethi Jyothi, IIT Bombay
6. Pabitra Mitra, IIT Kharagpur
7. Chetan Arora, IIT Delhi
8. P J Narayanan, IIIT Hyderabad
9. C. V. Jawahar, IIIT Hyderabad

****

PCC CS-403	Operating Systems	3L:0T:4P	5 Credits

Prerequisites details:

1. Familiarity with the C/C++ programming language
2. PCC 402: Computer Organization and Computer Architecture
3. CS403 to be scheduled in semester after CS402
   (both courses not to be scheduled in the same semester)

Learning Outcomes of the course (i.e. statements on students' understanding and skills at the end of the course the student shall have):

Essential:

1. To understand the role, functionality and layering of the systems software components
2. To understand the design and usage of the OS API and OS mechanisms
3. To understand the details of the abstractions and interfaces provided by the OS for program execution and execution requirements --- processes, threads, memory management, files.
4. To understand problems arising due to concurrency and related synchronization based solutions.
5. Hands-on and practical experience with usage of the OS API and basics of OS mechanisms

Desirable/Advanced:

1. To gain an in-depth understanding of the design and implementation of OS internals via a teaching OS
2. To be able to implement incremental changes to the functionality of a teaching OS

Detailed contents for Essential Learning Outcomes:

Module (appx duration in weeks)	**Topics**	**Pedagogy / teaching suggestions**	**Nature of lab / assignment / practice**
Module 1: Introduction to Operating Systems (0.5 week)	Application requirements The systems stack and role of OS, resources, abstractions and interfaces Components overview of an OS Examples of different types of OSes (RTOS vs. desktop vs. mobile etc.), OS and OS distributions.	T1: Chapter 2 T2: Chapters 1, 2 T3: Chapter 1 R1: Chapter 1 R2: Chapters 1, 2 R4: Chapter 1 R7:Chapter 1 R8: Chapters 1 to 8	1. Usage of tools --- unix shell commands (file commands, ps, ls, top), text editor (nano, vi, gedit, emacs) 2. C programming language refresher --- header files, compilation and linking using GCC, program execution, functions,

		O1: Bash Guide for Beginners	argument passing, structures, pointers, file handling.
Module 2: Computer organization and computer architecture refresher (0.5 week)	Basic organization of hardware components Role of OS relative to hardware functionality with examples related to the von Neumann architecture	T2: Chapter 1 T3: Chapter 1 R1: Chapter 1 R2: Chapters 1, 2 O1: Bash Guide for Beginners	1. Usage of tools --- GCC, GDB, Objdump, shell scripts
Module 3: Processes (1.5 weeks)	Process abstraction --- program vs. process, Process Control Block (PCB) Design of system calls --- invocation and basic OS handling Process control system calls --- fork, wait, exec, getpid, getppid and variants The limited direct execution model	T1: Chapters 4,5,6 T2: Chapters 6, 7 T3: Chapters 7, 8	1. Simple strace usage to showcase different interfaces (stdlib, system call) 2. Tools usage --- ps, pstree, top 3. Usage of process control system calls to identity process identifiers, create process hierarchies, launch new executables, control exit sequence of parent and child processes. 4. Familiarity with files in the /proc/<pid>/ directory
Module 4: Memory management (3 weeks)	Address bus and memory access Memory view of a process --- heap, stack, code, data Process memory usage requirements The address space abstraction using virtual memory and related system calls (mmap, munmap, sbrk, mprotect) Address translation mechanisms --- static	T1: Chapters 13,14,15,16, 17,18,19, 20 T2: Chapter 9 R8: Chapters 5,6	1. (Virtual) addresses of variables and initialized pointers. 2. Use of malloc() and demonstration of per-process virtual addresses 3. Tools usage --- strace, free, top, htop, vmstat, /proc/<pid>/maps 4. Free memory statistics correlated with malloc(). Number of system calls and malloc() usage.

	mapping, segmentation, paging Page faults, page sharing, read/write permissions, swapping, process vs. OS memory Memory bookkeeping and management --- motivation and mechanisms (process and OS) Case studies --- (i) malloc and (ii) role of OS for program to process		5. Implement a custom memory allocator using system calls
Module 5: Process management (1.5 weeks)	The process lifecycle--- source code to execution The OS mode of execution --- limited direct execution recap, interrupts, system calls The process context switch mechanism and PCB state Scheduling policies --- set of scheduling metrics, goals and examples (interactive vs. real-time, priority)	T1: Chapters 7,8 T2: Chapter 8	1. User mode programs to demonstrate LDE 2. Demonstration of process execution interleaving in different orders 3. Simulation based analysis of scheduling policies 4. Tools usage --- nice, /proc/<pid>/status
Module 6: Concurrency and Synchronization (3 weeks)	Motivation --- application, process and OS use cases. Introduction to threads and the pthread API Synchronization primitives --- limitations of software solutions, atomic instructions, test-and-set, spinlocks, mutexes, condition variables, semaphores	T1: Chapters 26, 27, 28, 29, 30, 31, 32 T3: Chapters 11, 12	1. Creation of threads using the pthread API and modification of shared variables with and without synchronization 2. Using spinlock, mutexes and condition variables to implement semaphores, barriers (using the pthreads API) 3. Implement solutions to the producer-consumer, reader-writers problems using the

	Introduction to the pthread synchronization API Case studies --- producer-consumer, reader-writers, barriers Discussion on issues with concurrency —race conditions, deadlocks, order violation.		different synchronization primitives 4. Develop synchronization solutions for applications that use shared data (e.g., ordering of threads, concurrent hash tables, etc.) 5. Optional: Using shared memory and semaphores implement synchronized access to a shared memory area across processes (e.g., a message queue).
Module 7: File systems (2 weeks)	Persistence and the File abstraction Hardware view: Hard disk architecture and its interfacing Process view: System calls for file handling Roles and responsibilities of file system File system design details--- file and file system metadata, directory structure, caching optimizations File System case study (the Unix file system etc.)	T1: Chapters 36,37,39, 40 T2: Chapters 4,5 T3: Chapter 3	1. Tools usage --- state, file, du, df, fsck. 2. Implementation of file utilities (e.g., find, grep) using the system call API. 3. Implement a simple file system to handle files on an emulated disk (via a large file) --- file system API, superblock, inode and data block management.

Detailed Contents for Desirable Learning Outcomes:

Module	**Topics**	**Pedagogy teaching suggestions**	**Nature of lab / assignment / practice**

1. Introduction to setup and usage of a teaching OS	Setup, configuration and usage of a teaching OS Understanding ISA details of the teaching OS Basics of interrupt handling, end-to-end execution of a system call and context switching mechanisms	T4	1. Configuration and setup of a teaching OS and writing of simple user mode programs (e.g., xv6) 2. Simple logging style modification to system calls (e.g., #times a system call was invoked, histogram of number of system call invocations, …) 3. Implement new system calls (e.g., read PCB elements to user space of a process) (xv6 based)
2. Basic modifications to the teaching OS functionality saved	OS context of execution --- address space, stack Address translation mechanism in the teaching OS The interrupt handling mechanism (context save and restore, privilege level change, interrupt dispatch) Example usage of locks in the teaching OS File system state in the operating system (file descriptors, file objects, inodes, page cache …) and in the teaching OS	T4	1. Write a system to call to output per process address space details 2. Write a system to call to determine physical address of a virtual address 3. Observe the stack pointers, privilege level registers in user and OS modes 4. Modifying/profiling behaviour of exception handlers 5. Observe process file table entries and file objects across parent and child processes
3. Advanced modifications to teaching OS	Context switch design and implementation in the teaching OS	T4	1. Write a system call to allocate the same physical block to

functionality	Lazy allocation, page fault management and swapping/demand paging Understand implementation of the synchronization primitives in the teaching OS		different virtual addresses 2. Implement lazy allocation of physical memory to processes 3. System call to print saved state of any process 4. Write a system call to induce page faults 5. Design and implement a shared message queue between processes to be used via the system call interface

Suggested text books / Online lectures or tutorials:

T1. Operating Systems: Three Easy Pieces
Remzi H. Arpaci-Dusseau and Andrea C. Arpaci-Dusseau
Arpaci-Dusseau Books, LLC
https://pages.cs.wisc.edu/~remzi/OSTEP/ (online version)

T2. Design of the UNIX Operating System
Maurice J. Bac
Pearson Education India; First edition

T3. Operating System Concepts
Ekta Walia
Khanna Publishing House; Second edition

T4. Advanced Programming in the UNIX® Environment
W. Richard Stevens, Stephen A. Rago
Pearson Education India; Third edition

T5. Xv6, a simple Unix-like teaching operating system
Frans Kaashoek, Robert Morris, and Russ Cox
[T4-R] https://github.com/mit-pdos/xv6-riscv (RISC-V version)

[T4-X] https://github.com/mit-pdos/xv6-public (x86 version)

Suggested Online content:

1. The Linux Documentation Project, www.tldp.org

Suggested reference books / Online resources:

R1. Modern Operating Systems, Andrew S. Tannenbaum and Herbert Bos, Pearson Education India; 4th edition

R2. Operating System Concepts, Avi Silberschatz, Peter Baer Galvin, Greg Gagne, Wiley India; 9th, edition

R3. Operating System courses offered on NPTEL, https://nptel.ac.in/

R4. Think OS, A Brief Introduction to Operating Systems. Allen B. Downey https://www.greenteapress.com/thinkos/index.html

R5. Linux Kernel Development, Robert Love, Pearson Education India; 3rd edition

R6. Operating Systems: Principles and Practice, Thomas Anderson, Michael Dahlin, Recursive Books; 2nd Edition, https://ospp.cs.washington.edu/index.htm

R7. Computer Systems: A Programmer's Perspective, Randall E. Bryant, David R.O' Hallaron, Pearson Education India; 3rd edition.

R8. The C Programming Language, Brian Kernighan, Dennis Ritchie, Pearson Education India; 2nd edition

Prepared by:

1. Purushottam Kulkarni, Department of Computer Science and Engineering (Indian Institute of Technology, Bombay)
2. Chester Rebeiro, Department of Computer Science and Engineering (Indian Institute of Technology, Madras)
3. Debadatta Mishra, Department of Computer Science and Engineering (Indian Institute of Technology, Kanpur)

The Discipline Graduate Attributes (GAs) to which this course contributes significantly: *CS3*

Other discipline GAs to which this course may contribute somewhat*: CS6, CS8*

****

MC	Constitution of India/ Essence of Indian Knowledge Tradition	-L:-T:-P	0 Credits

Course Content

Unit 1: **The Constitution - Introduction**

- The History of the Making of the Indian Constitution
- Preamble and the Basic Structure, and its interpretation
- Fundamental Rights and Duties and their interpretation
- State Policy Principles

Unit 2 – Union Government

- Structure of the Indian Union
- President – Role and Power
- Prime Minister and Council of Ministers
- Lok Sabha and Rajya Sabha

Unit 3 – State Government

- Governor – Role and Power
- Chief Minister and Council of Ministers
- State Secretariat

Unit 4 – Local Administration

- District Administration
- Municipal Corporation
- Zila Panchayat

Unit 5 – Election Commission

a. Role and Functioning
b. Chief Election Commissioner
c. State Election Commission

Suggested Learning Resources:

S. No.	Title of Book	Author	Publication
1	Ethics and Politics of the Indian Constitution	Rajeev Bhargava	Oxford University Press, New Delhi, 2008
2	The Constitution of India	B.L. Fadia	Sahitya Bhawan; New edition (2017)
3	Introduction to the Constitution of India	DD Basu	Lexis Nexis; Twenty-Third 2018 edition

Suggested Software/Learning Websites:

1. https://www.constitution.org/cons/india/const.html
2. http://www.legislative.gov.in/constitution-of-india
3. https://www.sci.gov.in/constitution
4. https://www.toppr.com/guides/civics/the-indian-constitution/the-constitution-of-india/

Alternative NPTEL/SWAYAM Course:

S. No.	NPTEL ID	NPTEL Course Name	Instructor	Host Institute
1	12910600	CONSTITUTION OF INDIA AND ENVIRONMENTAL GOVERNANCE: ADMINISTRATIVE AND ADJUDICATORY PROCESS	PROF. SAIRAM BHAT, PROF. M. K. RAMESH	NATIONAL LAW SCHOOL OF INDIA UNIVERSITY

Suggested text books:

T1. Introduction to Indian Knowledge System: Concepts and Applications, authored by Dr. B. Mahadevan, Professor of IIM Bangalore and founding Vice Chancellor of Chinmaya Vishwa Vidyapeeth; Dr. Vinayak Rajat Bhat, Associate Professor, Chanakya University, Bengaluru; and Dr. Nagendra Pavana R.N., Faculty at the School of Vedic Knowledge Systems, Chinmaya Vishwa Vidyapeeth; Forewords by Dr. Anil Sahasrabudhe, Former Chairman AICTE; Prof. Subhash Kak, Oklahoma State University, USA; and Dr. S. Sadagopan, Chairman, BoG, IIITDM – Kancheepuram & Founder Director, IIIT-Bangalore.

****

SEMESTER – VI

SEMESTER VI

PCC CS-601	Computer Networks	3L:0T:4P	5 Credits

Pedagogy and Learning Outcomes

This course introduces students to the fundamental principles of computer networks, and to their use in the Internet. Lab assignments cover network programming, network tools, applications, and simulation. Through these assignments, students get a broad understanding by building, operating and tuning components of the Internet. The top-down, application-driven design of the course motivates students with familiar uses and problems of their digital world and enables them to work with real-world applications from early in the course.

At the end of the course the student shall:

***Essential*:**

1. Understand the architecture principles that have enabled the orders of magnitude expansion of the Internet
2. Understand networked applications and their protocols, their installation, operation and performance tuning
3. Understand layering as a means of tackling complexity, layering applied to the Internet
4. Understand protocols as a structured means of reliable communications
5. Be conversant with network programming using the socket API
6. Be familiar with tools for configuring, monitoring and tuning the Internet and hosts

***Desirable/Advanced*:**

1. Understand basics of data center networks and software-defined networks (SDN)
2. Understand cellular networks, mobility and the impact on applications
3. Understand the interaction between streaming media applications and the underlying network infrastructure.

Detailed contents for Essential Learning Outcomes:

Module (appx duration in weeks)	**Topics**	**Pedagogy / teaching suggestions**	**Lab / assignment / practice**
Module 1: Introduction to the Internet (1 week)	Overview of how the Internet works. Understand at a high level what happens when we browse a website. Understand basic terminology like browser, web server, URL, domain name, IP address, packets. (1 hour) Overview of the design principles of the Internet: packet switching vs circuit switching, store-and-forward networks, layering for modularity. Introduction to the various layers in the Internet. (1 hour) Introduction to performance metrics like end-to-end throughput, delay, jitter and drop rates in a network. Statement of Little's Law. How performance is measured. (1 hour)	**Kurose & Ross**: Chap 1 (sec. 1.1-1.5). **Raj Jain**: Sec. 3.2 - 3.4 Top-down sequence is recommended. Performance metrics and measurement concepts are introduced in Module 1 and used in all modules (lectures and labs) as appropriate.	- Use Linux tools like ifconfig, dig, ethtool, route, netstat, nslookup, and ip to understand the networking configuration of the computer that the student is working on. - Use Wireshark to capture packets when browsing the Internet. Examine the structure of packets: the various layers, protocols, headers, payload. This resource is useful for many of the lab assignments: **https://gaia.cs.umass.edu/kurose_ross/about.php**

Module (appx duration in weeks)	**Topics**	**Pedagogy / teaching suggestions**	**Lab / assignment / practice**
Module 2: Application layer (2 weeks)	- Internet names, how DNS works. (1 hour) - Application layer protocols: HTTP, SMTP, SNMP, web applications. (3 hours) - Peer-to-peer applications. P2P file distribution. (1 hour) - Audio and video streaming. Challenges of streaming over best effort IP. (1 hour)	**Kurose and Ross**: Chap 2 (sec. 2.1-2.5), Sec. 7.1.3	- Install and configure some network applications, e.g. Apache, Bind (DNS), etc. - Understand various header fields and their usage in different application layer protocols using Wireshark packet capture.
Module 3: Linux Network Programming (0.3 week)	- Introduction to socket programming in Linux. Understand how to build a simple client-server application using TCP/UDP sockets. (1 hour)	**Kurose and Ross**: Sec 2.7.	- Socket programming: write a simple client-server program using TCP and UDP sockets. **Optional**: Modify server to handle multiple clients concurrently.

Module (appx duration in weeks)	**Topics**	**Pedagogy / teaching suggestions**	**Lab / assignment / practice**
Module 4: Transport Layer (2.3 weeks)	- Importance of the transport layer; end-to-end principle. Transport layer protocols: basics of TCP and UDP, process-to-process delivery, multiplexing, port numbers, header structure. (2 hours) - Reliable transmission of packets over an unreliable network: sequence numbers, ACKs, timeout, retransmissions. Stop and wait, and sliding window. (2 hours) - TCP connection setup and teardown. (1 hour) - Flow control and congestion control at the transport layer. Differences between the two. Overview of TCP congestion control: Slow start and reaction to timeouts (2 hours) - TCP congestion control: Slow start; congestion avoidance using loss-based and delay-based control. (1 hour)	**Kurose & Ross**: Chap. 3 (sec. 3.1-3.6) **Lab**: Optional -- https://www.isi.edu/nsnam/ns/tutorial/nsscript1.html#second Validation tests and demos: https://www.isi.edu/nsnam/ns/ns-tests.html	- Measure TCP throughput between two hosts in a network using tools like iperf. Modify TCP configuration parameters. Use the **tc** Linux utility or similar to control bandwidth, delay, loss. Observe impact on measured throughput. - Experiment with multiple applications running concurrently to generate congestion. **Optional**: Observe the behaviour of congestion control protocols in ns-2/ns-3, change various network parameters and observe evolution of the TCP congestion window.

Module (appx duration in weeks)	**Topics**	**Pedagogy / teaching suggestions**	**Lab / assignment / practice**
Module 5: The IP Layer (2 weeks)	**[A] Network architecture and performance** - Network topology; Router architecture: queueing and switching. (2 hour) - Performance evaluation of a network link: traffic characteristics, performance measures, Kendall's notation. (1 hour) **[B] IP Protocol** - Need for an Internet address, and its design. Hierarchical IP addressing, IPv4 and IPv6, structure of IP datagram, IP forwarding. - NATs, security attacks and defences: DMZ, firewalls. (3 hours)	**Kurose & Ross**: Chap. 1.4; Chap. 4 (Sec. 4.1, 4.3, 4.4) **S. Bose:** Chap. 1; Chap. 2.3	- Use tools like ping and traceroute to explore various Internet paths to popular servers. - Use web-based tools like the **whois** utility to query Internet registries, and understand which IP addresses are allocated to the student's network. Find out which are the major ISPs, and which is the ISP of the student's network.
Module 6: Routing protocols and Internet architecture (2 weeks)	- Routing protocols: Link state routing. Distance vector routing: count-to-infinity, routing convergence. (3 hours) - Understand the structure of the Internet: end-user organizations and ISPs. Understand the difference between intra-domain (OSPF) and inter-domain (BGP) routing. Intra-domain routing: OSPF. (3 hours)	**Kurose & Ross**: Chap. 4 (Sec. 4.5 - 4.6)	- Configure a simple mesh network using computers in the lab, or using Mininet. Setup static routes to conform to the desired mesh topology. - Use NS-2/NS-3 to simulate a mesh of at least 4 nodes and 3 links to evaluate performance under various conditions

Module (appx duration in weeks)	**Topics**	**Pedagogy / teaching suggestions**	**Lab / assignment / practice**
Module 7: Data Link Layer (2 weeks)	- Mechanisms for error detection/recovery: Parity checks, CRC (1 hour) - Medium access protocols: Polling vs. contention-based: TDM, Aloha, CSMA/CD (3 hours) - Switched LANs: L2 addressing and ARP, Ethernet frame structure, learning switches. (2 hours)	**Kurose & Ross**: Chap. 5 (Sec. 5.1 - 5.4)	- Use Linux network tools like ethtool to observe and analyze link layer packet statistics and errors. **Optional**: Use NS-2/NS-3 to simulate medium access protocols. Observe contention, collisions and packet loss in medium access protocols. Observe the working of error detection/recovery mechanisms.
Module 8: Wireless Networks (1 week)	- Wireless physical layer: signal-to-noise ratio, bit error rate, modulation, multipath, interference (1 hour) - Wireless LANs: 802.11 architecture (access points, SSID, channels, beacons, scanning, association), 802.11 CSMA-CA protocol; summary of 802.11 variants (3 hours)	**Kurose & Ross**: Chap 6 (Sec. 6.1, 6.2, 6.3)	- Use cellphone to measure Wi-Fi signal strength (RSS) at various places in the campus. Draw a contour map with access points and RSS levels. Correlate with upload/download speed using tools like Measurement Lab speed test. **Optional**: Understand the behavior of Wi-Fi using NS-2/NS-3.

Detailed Contents for Desirable Learning Outcomes:

Module (appx duration in weeks)	Topics	Pedagogy/ teaching suggestions	Lab / assignment / practice
Module D1: SDN and data center networking	Data center network design and topology. Transport protocols optimized for data centers. Introduction to software defined networking.	**Kurose and Ross:** Chap. 5.6 Refs [5], [6]	Simulate transport protocols optimized for data centers in NS-2/NS-3.
Module D2: Mobile data networks	Cellular Internet access. Mobility management and handovers. Impact of mobility on higher layers.	**Kurose and Ross:** Chap. 6.4-6.8	Use cellphone to measure cellular signal strength (RSS) at various places in the campus. Draw a contour map with cellphone towers and RSS levels. Correlate with upload/download speed using tools like Measurement Lab speed test.
Module D3: Streaming media protocols and services	Multimedia applications. Audio and video streaming over UDP, HTTP. Adaptive streaming. Voice over IP. Recovering from packet loss and jitter. QoS. Protocols for real time applications.	**Kurose and Ross:** Chap. 7	Implement a streaming audio/video server using open-source software.

Suggested text books / Online lectures or tutorials:

1. J.F. Kurose and K.F. Ross, *Computer networking: a top-down approach*, 6th edition, Pearson, 2017. (*6th edition is low-cost Indian edition. 7th edition is high-cost, may be used if available.)*
2. Bhavneet Sidhu, An Integrated Approach to Computer Networks, Khanna Publishing House, 2021.

Suggested reference books / Online resources:

1. R. Jain, *The art of computer systems performance analysis,* Wiley India, 1991.
2. S.K. Bose, *An Introduction to Queueing Systems*, Springer Science + Business Media New York, 2012.
3. A.S. Tanenbaum and D.J. Wetherall, *Computer Networks*, 5th edition, Pearson, 2013.
4. Larry Peterson and Bruce Davie, *Computer Networks: A Systems Approach*, 6th Edition,

available at https://book.systemsapproach.org/

- N. Feamster, J. Rexford, Ellen Zegura, "The Road to SDN", *ACM Queue*, 2013.
- Alizadeh et al., "Data Center TCP", *ACM SIGCOMM 2010*.
- *The Network Simulator - ns-2*, https://www.isi.edu/nsnam/ns/
- E. Altman and T. Jimenez, *NS2 Simulator for Beginners*, 2003.
- *ns-3 network simulator*. https://www.nsnam.org/

Prepared by (alphabetical order)**:**

1. Timothy A Gonsalves, School of Computing & Electrical Engg, IIT Mandi
2. BN Jain, Department of Computer Science & Engg, IIIT Delhi
3. Vinay Ribeiro, Department of Computer Science & Engg, IIT Bombay
4. PM Sreelakshmi, School of Computing & Electrical Engg, IIT Mandi
5. Mythili Vutukuru, Department of Computer Science & Engg, IIT Bombay

The Discipline Graduate Attributes (GAs) to which this course contributes significantly: CS3

Other discipline GAs to which this course may contribute somewhat: CS4, CS6, CS8

******

PEC CS-601	Introductory Cyber Security	3L:0T:4P	5 Credits

Learning Outcomes of the course (i.e. statements on students' understanding and skills at the end of the course the student shall have):

Having completed this course, a student should be able to:

Essential:

1. Understand the importance of cyber security (data confidentiality, Integrity, and Availability) and various recent attacks on important digital systems such as banking, e-commerce systems, e-governance systems etc.
2. Understand basic cryptography concepts – symmetric vs asymmetric cryptography, Public Key Crypto Infrastructure (PKI), Symmetric Ciphers, Hashing, Digital Signatures.
3. Understand methods and tools for authentication, authorization, privilege, and their needs in securing an organization's IT system.
4. Understand the common vulnerabilities in applications, web applications, network, and the Internet Infrastructure.
5. Understand the methods and tools for Intrusion Detection (network and host intrusion detection) and perimeter security (firewall).
6. Understand basic malware functions and indicators of compromise.

Desirable/Advanced:

1. Understand basic mobile application security issues and android platform architecture for securing app execution.
2. Understand wireless LAN security issues.

Detailed contents for Essential Learning Outcomes:

Module (appx duration in weeks)	Topics	Pedagogy / teaching suggestions	Nature of lab / assignment / practice
Module 1: Introduction and basic terminology (~1 weeks)	Cyber Security and CIA Triad, basic cyber threats to CIA, cyber-attack surfaces, recent cyber-security incidents and their high-level analysis	Example Driven Lectures with examples drawn from most recent incidents	None
Module 2: Basic Cryptography (1 - 2 weeks)	Role of Cryptography in ensuring confidentiality for data at rest, data in motion, and data in process.	Provide good intuition than nitty gritty of algorithms, or going through	Using library functions to use RSA, AES, SHA-256 and show the result of

	Symmetric and Asymmetric Cryptography, their needs as complementary of each other, some basic symmetric and asymmetric algorithm outlines (RSA, DH, DES, AES) Role of cryptography in data integrity, non-repudiation Hashing and Digital Signature and some example hash function outlines (MD5, SHA-256), understanding digital signature and its role. Digital Certificate and PKI. Importance of the role of a proper Pseudo Random Number Generator	ciphers in details. The intuition behind PKI, Digital Certificate.	encryption, Hashing etc. Taking apart a digital certificate and show the various components and their significance
Module 3: Authentication, Authorization and Privilege (1 week)	Importance of strong Authentication, distinction between authorization and authorization, importance of authorization, access control, Mandatory and Discretionary Access control, role based authorization, privilege and privilege escalation	Intuition of distinguishing between authorization from authentication, access control lists, MAC vs DAC with examples, importance of distinct privileges, principle of least privilege, show example of privilege escalation	Lab on 2 factor authentication, Lab on privilege escalation example
Module 4: Application Security (4-5 weeks)	Basic application vulnerabilities (Buffer overflow, Integer Overflow, format string vulnerability), Basic mitigations of buffer overflow – platform bases, compiler based, secure programming practice Web Client Security, Same Origin Principle, DOM, Java Script	Intuition of why these vulnerabilities happen, and how various mitigation techniques have been developed, why the	Lab1: Buffer overflow, integer overflow and format string vulnerability testing in vulnerable applications

	Vulnerability, Cookies and Cookie Attributes Secure, http only, Concept of session and session ID, Session hijacking vulnerability, http vs. https and SSL/TLS and version issue Web Server Security – XSS, CSRF, SQL Injection, Command Injection concepts, examples of each and mitigation techniques Vulnerabilities in DNS, Routing and IP protocols especially in IPv4 and suggested remedies with DNSSEC, S-BGP, and IPSec	mitigation techniques are not enough and can be escaped by determined attackers, and examples from real attacks	Lab 2: DVWA based command injection. SQL injection, XSS and CSRF
Module 5: Perimeter protection and Intrusion Detection (2 weeks)	Host Intrusion Detection techniques, what are the indicators to look for and how an SIEM tool can consolidate such indicators into a management console Network Intrusion Detection – signature based vs. behavior based, Snort Firewall vs. Intrusion Detection tool, Firewall rules and customization techniques	Intuition about indicators that could indicate a host as compromised, and how multiple hosts/endpoints can be monitored Intuition behind signature vs. behavior based network monitoring and detection of intrusion Hands on with Snort installation	Lab1: Students are asked to install Wazuh and monitor a host Lab 2: Students are asked to install snort and monitor a network on their local network
Module 6: Basic Malware Analysis (1 week)	Various malware classes and their characteristics Difference between static analysis and dynamic analysis Signature vs. behavioral detection techniques	Intuition about various malware classes and their modus operandi, discuss why static and dynamic analysis complement	Lab 1: US static analysis tools to find how an executable can be analyzed.

		each other, what kind of information is obtainable from static vs. dynamic analysis, demonstrate some static analysis tools	

Detailed Contents for Desirable Learning Outcomes (optional, <= 3 modules):

Module	Topics	Pedagogy teaching suggestions	Nature of lab / assignment / practice
Module 7: Mobile Application Security (1 week)	Basic mobile attack surface and the ideas of permissions, and their abuse Execution model of mobile apps in Android (Sandboxing) and communication	Provide intuition on mobile malware and how they work, give example of mobile malware attacks, provide intuition of execution model of Android and demonstrate Mandatory Access Control idea in action, SE Linux being part of Android	None
Module 8: WLAN Security (1 week)	Some common ways WLAN are compromised including weak cipher such as WEP, evil twin attack, unauthorized access point based attacks (rogue WLAN) etc.	Provide students idea about how to look for signs of these rogue WLAN, evil twins, public Wi-Fi etc.	None

Suggested text books / Online lectures or tutorials:

1. Ross J. Anderson, Security Engineering, Third Edition, Wiley, Nov 2020
2. Cyber Crime and its Prevention in Easy Steps, Debtoru Chatterjee, Khanna Publishing House, 2022.
3. Cyber Attacks and Counter-Measures Made Simple, Debtoru Chatterjee, Khanna Publishing House, 2022

Suggested reference books / Online resources:

1. The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws 2nd Edition by D Stuttard and M Pinto
2. Cryptography and Network Security by William Stallings.

3. The Hacker Playbook: Practical Guide to Penetration Testing (vol. 1 and 2) by Peter Kim.
4. Introduction to Security of Cyber-Physical Systems, Jeeva Jose, Khanna Publishing.
5. Mastering Hacking, Harsh Bothra, Khanna Book Publishing House.

Prepared by: Sandeep K. Shukla (IITK), Vinod Ganapathy (IISc), Sambuddho Chakravarty (IIITD), S. Venkatesan (IIITA).

The Discipline Graduate Attributes (GAs) to which this course contributes significantly: *CS1, CS3, CS6*

*****

PCC CS-504	Theory of Computation	3L:1T:0P	4 Credits

Essential Learning Objectives:

1. Understand models and abstractions: automata as a basic model of computation
2. Link between languages, automata, and decision problems.
3. How to build new models from old ones: product, union, closure properties.
4. Argue about limitations of computational models.
5. Understand algebraic formalisms of languages such as regular expressions, context-free grammar.
6. Understand algorithms and computability through the lens of Turing machines.
7. Existence of unsolvable problems and what that means.
8. Relations between the various computational models.

Advanced/Desirable Learning Outcomes: Nil

Module (appx duration in weeks)	**Topics**	**Teaching Suggestions**	**Learning outcomes**
Module 1: Finite Automaton (4-5 weeks)	-Why automata theory? -Alphabets, formal languages, and problems. -What are regular languages and automata models for them: Deterministic Finite automaton, Formal argument of correctness, Regular languages	-Sec 1.1 of T2 -Sec 1.2, 1.5 of T2 -Sec 1.1 of T1	F. Familiarity with notations. U. Give examples of languages, regular languages. U. Design finite automata, both deterministic and nondeterministic for a given language.

	-Properties of regular languages - Closure, properties, product construction -Limitations of Automata Non-regularity, Pumping Lemma -Non-Deterministic Finite Automaton, Subset construction, Equivalence with DFAs. -Regular expressions. Equivalence with regular languages. -Algorithms for regular languages, Minimization and its algorithm. -(suggested) Myhill-Nerode relations, Characterization of regular languages	-Sec 1.1 of T1 -Sec 1.4 of T1 -Sec 1.2 of T1 -Sec 1.3 of T1 -Sec 4.3, 4.4 of T2 -Lecture 15,16 of R1 +applications of automata to text search and NLP +applications of regular expressions for text search in UNIX. **Advanced Topics:** - 2DFAs, Equivalence with DFAs using Myhill-Nerode Relations (Lecture 17,18 of R1)	R. Write formal proof of correctness of a DFA U. Give examples of non-regular languages and prove that language is non-regular using pumping lemma F. Understand the difference between determinism and nondeterminism U. Use closure properties to show non-regularity U. Design regular expressions U. Use the minimization algorithm to minimize a given DFA U. (suggested) Apply Myhill-Nerode Theorem to show that a language is regular or non-regular

Module 2: Grammars, Context-free Languages and machine models. (4-5 weeks)	-Grammars and the motivation from language theory. -Context-free grammars, closure properties. Chomsky Normal Form for CFGs. -PDAs. Empty-stack vs Final state acceptance conditions. Equivalence of PDAs and CFGs. -Limitations of PDA computation, non-context-free language. Pumping Lemma for CFLs. -Deterministic CFLs and PDAs. -(suggested) CYK Algorithm for parsing of CFLs.	-Sec 2.1 of T1 -Sec 2.1 of T1 -Sec 2.2 of T1 -Sec 2.3 of T1 -Sec 2.4 of T1 -Sec 7.4 of T2 +applications to parsers and compilers. **Advanced Topics:** - Ogden's Lemma.	U. Design CFGs and PDAs for CFLs R. Prove correctness of CFGs F. Understand that regular languages are a subset of CFLs. R. Prove equivalence of CFGs and PDAs U. Argue a language is non-CFL using pumping lemma F. Familiarity with DPDAs U.(suggested) Construction of DPDAs U. (suggested) Parsing using CYK algorithm

Module 3: Turing machines and Computability . (4-5 weeks)	-Modeling computation using Turing Machines. Equivalent models. Church Turing Hypothesis. -Decidability and Turing recognizability (i.e., recursive and recursively enumerable). Closure properties. -Undecidability by diagonalization. -Reductions to show undecidability. Examples of reductions. -Resource bounded Turing machines & Intro to Complexity. Basic complexity classes. Time bounded classes: P, NP, EXP. -(suggested) Post's correspondence problem and other undecidable problems -(suggested) Polytime reductions, NP-completeness, Cook-Levin Theorem without proof	-Sec 3.1, 3.2, 3.3 of T1 -Sec 4.1 of T1 -Sec 4.2 of T1 -Sec 9.3 of T2. Sec 5.1, 5.3 of T1. -Sec 7.1 of T1 -Sec 5.2 of T1 -Sec 7.3, 7.4, 7.5 **Advanced Topics:** - Rice's Theorem - Space bounded computations and complexity, PSPACE	F. Understand relation between the various classes such as decidable, Turing recognizable., co-Turing recognizable. F. Give examples of decidable languages, undecidable languages, Turing recognizable languages. U. Prove a language is undecidable by reduction from a known undecidable problem F. Relation between basic complexity classes F. (suggested) Scenarios in which the reductions are used R. (suggested) Proving languages are NP-complete using reductions

Notations:

- Topic Categorization:
  Compulsory - Topics that should be covered.
  Suggested - Optional topics that the instructor can choose from given availability of time.

Advanced - Advanced topics in each module that an instructor can teach depending on the interest of the class.

- \+ indicates applications that could be mentioned in the class.
- Learning Outcome Categorization:
  ***F**amiliarity* - Student should be able to identify and comprehend *what* the topic is about. This corresponds to the cognitive levels of *knowledge* and *comprehension* of Bloom's taxonomy (see e.g., https://en.wikipedia.org/wiki/Bloom's_taxonomy).
  ***U**sability* - Student should be able to understand *how* a particular idea/topic can be used, to solve problems, design examples, etc. This corresponds to the cognitive levels of *application* and *synthesis* of Bloom's taxonomy.

  ***R**easoning* - Student should have a deeper understanding of a particular concept and *why* it works. This corresponds to the cognitive levels of *analysis* and *synthesis* of Bloom's taxonomy.

Nature of lab / assignment / practice / tutorial:

1. Make assignments using the books. To test:
   - what was done in the class
   - whether the student can think and apply the concepts.

2. Tutorials: Weekly problem-solving sessions.

Suggested textbooks:

1. Theory of Computation, R.B. Patel, Khanna Book Publishing, 2020.
2. Introduction to the Theory of Computation, 3rd edition. Michael Sipser, Cengage Publications (Low-cost Indian edition available).
3. Introduction to Automata, Theory, Languages and Computation. Third Edition. John Hopcroft, Rajeev Motwani, Jeffrey D. Ullmann, Pearson Publications (Low-cost Indian edition available).

Additional Reference Material:

1. Automata and Computability, Dexter C. Kozen. Part of the Undergraduate Texts in Computer Science book series (UTCS), Springer.
2. Elements of the Theory of Computation, 2nd edition. Harry Lewis, Christos Papadimitriou, Prentice Hall.

Prepared by: Jayalal Sarma, IIT Madras. S Akshay, IIT Bombay. Raghunath Tewari, IIT Kanpur.

The Discipline Graduate Attributes (GAs) to which this course contributes significantly: CS4

1. ***Other discipline GAs to which this course may contribute somewhat***: CS2

****

SEMESTER – VII

SEMESTER VII

PEC CS-602	Compiler Design /Construction	3L:0T: 4P	5 credits	Pre-Requisites: Data Structures

Prerequisite: C/C++/Java programming language. Data structures and algorithms. Automata theory.

Learning Outcomes,

Essential:

1. To understand the role, functionality and structure of program translation and interpretation in software development.
2. To understand the difference between abstraction levels of a high level language and a machine language.
3. To understand the role of a sequence of intermediate representations in lowering the level of abstractions in the process of language translation.
4. To get a first-hand experience of a practical application of elegant data structures, algorithms, and other core CS concepts such as automata theory.
5. To make effective use of tools such as LEX and YACC.
6. To become a much better programmer by appreciating all that happens behind the scenes in making an HLL program run.

Desirable/Advanced:

1. To understand some of the critical aspects of machine code generation.
2. To understand the issues in efficient code generation.

Syllabus for Essential Learning Outcomes:

Module (appx duration in weeks)	Topics	Pedagogy / teaching suggestions	Nature of lab / assignment / practice
Module 1: Introduction to Compilers (1 week)	Comparing abstractions of a high level language and a low level language; compilation as a series of steps for lowering the abstraction level through stepwise refinement; phases of compilation; bootstrapping; cross-compilation.	Sections 1.1 and 1.2 from the textbook, online resources [2] and [3].	Viewing the intermediate representations and the final assembly code generated by GCC/LLVM, relating them to the input program

			Programming Assignment #A0.
Module 2: Lexical Analysis (2 weeks)	The role of lexical analysis; Token, lexemes, and token codes; Regular Expressions (RE) to represent tokens, Deterministic finite automata (DFA),Traversing a DFA for recognising tokens; Generating a lexical analyzer using LEX/Flex.	Sections 3.1, 3,2, 3.4.4, 3.5, 3.6, 3.8.1, 3.8.3 from the textbook.	Writing LEX specifications and generating tokens for a given language, Programming Assignment #A1.
Module 3: Syntax Analysis (3 weeks)	Context Free Grammars (CFG), Concept of parsing, sentences and sentential forms, leftmost and rightmost derivations, parse trees, ambiguous grammars; Overview of top-down and bottom-up parsing; Option1: Introduction to shift reduce parsing; viable prefixes and valid items, Constructing LR(0) sets of items; Constructing SLR parsing tables; Generating a parser using a parser generator such as YACC/Bison. Option 2: Top-down parsing, Left factoring, Elimination of left-recursion, predictive parsing, recursive descent parsing, LL(1) parsing. Generating a parser using a parser generator such as ANTLR, JavaCC, etc.	This module is driven by the chosen parser generator. If YACC/Bison is chosen, the module should cover bottom up parsing. If Antlr or JavaCC is used, the module should cover top down parsing. Sections 4.1, 4.2, 4.3 from the textbook. Option1: Sections 4.5, 4.6 from the textbook. Option2: Section 4.4 from the textbook.	Writing YACC specifications, generating a parser by using the scanner generated in module 2. Precise error reporting using yytext and yylineno. Using a command line switch to optionally print token details. Programming assignment #A2.
Module 4: Semantic Analysis	The need of semantic analysis; abstract syntax trees for expressions, assignment statements and control flow	Sections 5.1, 5.2, 5.3, 5.4, and 6.1 from the textbook.	Writing a type checker to ensure that a syntactically correct

(1 week)	statements; attribute evaluation, syntax directed translation schemes (STDS);		MMC program is type-safe. Programming Assignment #A3.
Module 5: Applications of Semantic Analysis (3 weeks)	Applications of SDTS for (a) declaration processing and type checking, (b) generating three-address code	Sections 6.2, 6.3, 6.4, 6.5.1, 6.5.2, 6.9 from the textbook.	Write a translator to translate a type-checked MMC program to equivalent three-address code. Programming Assignment #A4.
Module 6: Run time support (1 week)	Parameter passing by value, reference, and name; activation records, stack and static allocation of activation records; translating a function call, allocating offsets to variables, generating code for function prologue, function epilogue, call sequence, and return sequence.	Sections 1.6.6, 7.1, 7.2.2, 7.2.3 from the textbook, Chapter 6.1 from the reference book and online material [4].	Not Applicable
Module 7: Introduction to Code Optimization (1 week)	Control flow graphs; Local optimizations (common subexpression, copy propagation, dead code elimination).	The focus of this module is on knowing the optimizations and not on the techniques of performing them. The desirable part would need an additional one week. Section 9.1 from the textbook..	Not Applicable
Module 8: Code Generation (2 weeks)	Generating assembly code from three address codes using simple register allocation and instruction selection.	By simple register allocation, we mean that all the values of temporaries are held in registers across three address code statements but not those of source variables.	

		Sections 8.1, 8.2, 8.6 in the textbook.	

Syllabus for Desirable/Advanced topics.

Module (appx duration in weeks)	Topics	Pedagogy / teaching suggestions	Nature of lab / assignment / practice
Module 2: Lexical Analysis (2 weeks)	4-arrays representation, observing the data structures in the scanner generated by LEX.	Section 3.9.8 from the textbook.	
Module 3: Syntax Analysis (3 weeks)	LR(1) and LALR(1) parsing and the option not chosen in the default offering	Section 4.7 from the textbook.	Error recovery using the error token in YACC and emitting meaningful error messages.
Module 7: Introduction to Code Optimization (1 week)	Global optimization (constant propagation, common subexpression elimination, copy propagation, dead code elimination, strength reduction)	Sections 9.2.5, 9.2.6, 9.4.1, 9.4.2, 9.4.3, 9.4.6, in the textbook.	
Module 8: Code Generation (2 weeks)	Register allocation using graph colouring, Optimal code generation for expression trees, Sethi Ullman algorithm, Aho Johnson algorithm.	Sections 8.8.1, 8.8.4, 8.10, 8.11 in the textbook.	Write a translator to translate code in 3-address-code form to assembly code. Programming Assignment #A5

Lab/Assignment Details

A0: Write 5 simple test-cases in MMC and then inspect the generated code.

A1: Write a lexer to recognize valid tokens.

A2: Write a parser to parse the given input MMC program.

A3: Write a type-checker for a syntactically correct input MMC program.

A4: Write a Translator that takes a type-checked MMC program and generates equivalent IR code in TACoC format; TACoC is a subset of MMC such that code is in a form similar to three address code. Details of TACoC is given towards the end of this section.

A5: Generate MIPS code. Use the SPIM simulator to run the code.

Suggested Pedagogy for the lab/assignments:

1. The instructor is expected to prepare a set of test-cases for assignments A1-A5.
2. For A1, A2, and A3 – the test cases should include some test-cases which throw lexical, syntactical and type errors, respectively.
3. For input test-case design the expected output and write a script to verify the output generated by the student assignment.
4. For each assignment, the student should use a script to build their code (using a Makefile/ant build script/shell script). The instructor must create some sample scripts and teach the same to the students.
5. The instructor should create a reference implementation for A1-A5.
6. The implementation can be done in Flex + Bison or Antlr or JavaCC.

Details of MMC and TACoC

MMC is a simple variant of C, and has many simplifications. This along with the simplified nature of TACoC ensures that the compiler can be written within a short span of time. Further details about this can be obtained from:

http://www.cse.iitm.ac.in/~krishna/aicte-compiler-design/Lab-Syllabus.docx

Texts and Other Material

1. [Text book] Aho, Lam, Sethi, and Ullman. Compilers: Principles, Techniques, and Tools. 2/e, Addison-Wesley, 2006.
2. [Reference book] Andrew Appel and Jens Palsberg. Modern Compiler Implementation in Java. 2/e, Cambridge University Press, 2002.
3. https://en.wikipedia.org/wiki/Cross_compiler.
4. https://en.wikipedia.org/wiki/Bootstrapping_(compilers)
5. https://en.wikipedia.org/wiki/Function_prologue_and_epilogue

****

Appendix

Appendix- 1

Professional Electives and Micro Specializations

Besides the core courses, programs normally have professional elective courses. Each HEI decides the electives it can or wishes to offer. In some areas may be desirable to organize a set of electives as micro-specializations. A *micro-specialization* is to provide a limited specialization in some sub-area of CSE, by offering suitable electives. The goal of the micro specialization is to provide deeper understanding and skill development in that area, and can provide multiple pathways to students, as different students can graduate with different specializations (or not). The areas in which micro specialization are offered should be aligned to industry careers or higher studies.

A micro specialization for CSE is defined as follows:

- It has a core course as the head (starting) course for the micro specialization
- It has a clearly defined goal, and learning outcomes for the goal
- It can have 2 +/- 0.5 additional courses (besides the head course) in the sub-area aligned to the goal. These courses can be full course (4-credits) or half-course (2 credit), and can be taken as electives by students (or extra credits.)

Based on the discussions and inputs, a few desired micro-specializations were identified. For a subset of these, a possible design of the micro-specialization is provided below. It should be pointed out that the list of courses in a specialization is illustrative – an Institution can replace or add a course aligned to the micro specialization goal. Institutions can also define a set of courses for a micro specialization and require that a subset be taken, with perhaps one being compulsory.

A special micro-specialization on **Advanced Mobile Communication / 5G** is also suggested by AICTE – further details on this are also provided later (in a slightly different format than others.)

It should be added that HEIs are completely free to decide whether to offer micro specializations or not, and if they decide to offer, which areas to provide the specialization in. How the micro specialization is to be reflected in a student's records/certificates is also to be decided entirely by HEIs based on their policies and practices.

Note that the list of additional courses mentioned in the micro specializations also provide a list of suggested professional electives. However**, there can be electives which are not a part of any specialization.**

1. Software Engineering Micro Specialization

Goal: The aim of this specialization is to provide an understanding of the importance and role of software engineering, an understanding of the important sub-areas of software engineering (SE), and to provide a deeper understanding of some of them.

Learning outcomes of the specialization:

1. Understand the role and importance of software engineering in developing industrial strength software, and the important tasks involved in engineering such software.
2. Ability to apply proper SE practices to develop in a team a working software system to solve some users' problem.
3. Understand and apply concepts in some sub-areas of SE like testing, maintenance, open source software, model based development, requirements engineering, etc.

Base Course for this specialization (core course): Advance Programming

Courses (Electives) Proposed for the Specialization:

(The courses for a micro specialization can be full course (4-credits) or half-course (2 credit). The total number of courses suggested is 2 +/- 0.5.) For this specialization, it is suggested that the first course should be required, followed by 2 half courses in some sub-areas of SE.

Course Name (full / half)	Purpose/Goal	Topics
Software Engineering (full)	To explain the iterative software development process and different aspects of it (e.g. short cycles, agile approaches, test-driven-development, etc.), and to apply the concepts to a team project to develop software	Iterative software development process, basic project planning for such a process, requirements including user interface, architecture and design, coding using modern IDEs, testing, integration and deployment
Software Testing (half)	Go deeper in software testing, including some modern tools	Unit testing and the test frameworks, test case design (incl. black box and white box), non-functional testing, test automation and tools, related metrics (e.g. coverage, performance, e.g.)
Open Source Software (half)	To expose to students open source software practices and ecosystem, and to expose them to using them	OSS evolution, current situation, general OSS practices, common OSS platforms and how to use them, small project on some OSS platform (e.g. GitHub)
Software Maintenance (half)	To explain the software maintenance cycle and different aspects of it, and develop capabilities in them	Defect cycle and bug management, regression testing, change management, refactoring, software evolution, related metrics, etc.

Prepared by: Pankaj Jalote (IIIT-Delhi), Raghu Reddy (IIIT Hyderabad), Vinay Kulkarni (TCS), Meenakshi Dsouza (IIIT Bangalore)

******

2. *Machine Learning Micro Specialization*

Goal: The aim of this specialization is to provide an understanding of the importance of machine learning in computer science, expose the advances in machine learning, and also to demonstrate the practical role of machine learning in related domains in artificial intelligence. This specialization also aims at providing deeper understanding of some of the sub areas, deeper analytical concepts, and the recent advances.

Learning outcomes of the specialization:

1. Appreciate the role of machine learning in computer science and data-driven problem solving.
2. Appreciate the role of recent and advanced machine learning algorithms and formulations in solving problems of practical importance. Expose students to principled ways of practicing machine learning in many unstructured real world problem settings.
3. Ability to apply principles of ML to develop practical solutions to problems seen around in industry, society and research, focusing on problem formulation, solution design, implementation and experiencing the empirical design process.
4. Providing practical experience in use of popular machine learning libraries and implementations. Get the students trained in the empirical science behind the design of machine learning based solutions.

Base Course for this specialization (core course):

1. Machine Learning

Courses (Electives) Proposed for the Specialization:

Note: The courses for the specialization can be full course (4-credits) or half-course (2 credit). The total number of courses suggested should be 2 +/- 0.5.

Course Name (full / half)	**Purpose/Goal**	**Topics**
Advanced Machine Learning (full)	- To introduce and be familiar with the popular deep learning architectures. - To expose the popular problem formulations beyond simple supervised learning used in situations of data scarcity, domain/distribution shift etc. using the popular ML/DL algorithms.	- CNNs, RNNs, Auto Encoders, Loss Functions and Training. - Transfer learning, domain adaptation, semi-supervised learning, active learning, self-supervised learning, incremental learning, few-shot learning Laboratory experiments to appreciate the utility

Computer Vision or Natural Language Processing (Full)	- To introduce a sub-area of Artificial Intelligence (perception) where machine learning formulations get extensively used. - Appreciate how ML has influenced the problems in this area, and how the type of data/problems in this area demanded newer solutions in ML.	- Formulations based on classification, regression, structured prediction in sub-tasks of CV/NLP. - Role of data and learning in Feature/Embedding. - Laboratory experiments to appreciate the utility - Encourage Course Project
Programming for Machine Learning (half; Lab Course)	- To expose students to popular ML libraries and frameworks such as PyTorch and Tensorflow). To develop empirical skills in design and implementation of ML solutions - To strengthen the hands-on skill of the student.	- Programming in popular libraries, Use of Cloud, GPUs, Debugging/Visualization. Working on large datasets. - Programming intensive. Use of popular (in industry, research) frameworks and tools expected.

Notes:

1. To strengthen the micro specialization, students may be encouraged to do the B Tech Project (or Internship or a summer project) in this area if the regulations (credit system) allow.
2. Second course may be adapted to the strength of the department as an “Applied Machine Learning Course " in an area of expertise in the department. However, the focus is on the use of ML formulations in this course.

Prepared by: Prof. C. V. Jawahar (IIIT Hyderabad), Prof. P. J. Narayanan (IIIT Hyderabad)

******

3. Distributed and Cloud Systems Micro Specialization

Goal: The aim this specialization is to provide: an understanding of the importance and role of distributed computing and data systems, an understanding of the important sub-areas of these systems such as cloud computing and big data platforms, a deeper understanding of their concepts, and the practical aspects of building applications for such systems. The aim of this specialization is to introduce important topics and themes of distributed systems and cloud computing, both from a theoretical and a practical perspective.

Learning outcomes of the specialization:

1. Understand the need for and the models of distributed systems, and ability to access remote applications and data over the network.
2. Understand key concepts for building scalable, reliable and consistent distributed applications.
3. Understand the design, implementation, deployment and use of distributed computing and data platforms.
4. Understand the models of Cloud computing, enabling technologies and how to build Cloud native applications

Base Course for this specialization (core course):

Computer Networks and/or Operating Systems

Courses (Electives) Proposed for the Specialization:

The courses for the specialization can be full course (4-credits) or half-course (2 credit). The total number of courses suggested should be 2 +/- 0.5.

Course Name (full / half)	Purpose/Goal	Topics
Principles of Distributed Systems (Half or Full)+	Understand the need for and the models of distributed systems, and ability to access remote applications and data over the network. Understand key concepts for building scalable, reliable and consistent distributed applications.	Need for distributed systems, models for coordination among multiple machines over the network. Models of distributed systems, remote invocation and remote storage. Concepts of performance, scalability, reliability, consistency and correctness.
Building Cloud and Big Data Applications (Full)	Understand the models of Cloud computing, enabling technologies and how to build Cloud native applications. Understand the design and use of distributed computing and data platforms.	Designing distributed applications/algorithms, Data-Intensive Computing and Data-Oriented Programming. Distributed execution and runtimes. Streaming data management and processing. Remote and Distributed file systems, Key Value Storage and NoSQL Columnar Store.

		Resource abstraction and Service oriented architecture, Computing Abstractions on the Cloud, and Building Cloud-native applications.
Cloud Systems Engineering (Half or Full)+	Understand the models of Cloud computing, enabling technologies and how to build Cloud native applications.	Cloud services models, Data center architecture and management. Virtualization and Containerization. Big data computing platforms, REST based web services.
Mini-Project/Hands-on Cloud Computing/Big Data (Half)*	Understand the design, implementation, deployment and use of cloud systems to help build distributed, scalable and reliable applications.	Hands-on mini programming project that makes use of the concepts and technologies learnt in the theory courses.

Notes:

1. Either separate or as a lab-component of one of the other theory courses
2. The course can be offered as half or full course depending on faculty availability.

Pedagogical References

1. Cristina L. Abad, Eduardo Ortiz-Holguin, and Edwin F. Boza. 2021. Have We Reached Consensus? An Analysis of Distributed Systems Syllabi. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education (SIGCSE '21)*. Association for Computing Machinery, New York, NY, USA, 1082–1088. DOI: https://doi.org/10.1145/3408877.3432409
2. Joshua Adams, Brian Hainey, Laurie White, Derek Foster, Narine Hall, Mark Hills, Sara Hooshangi, Karthik Kuber, Sajid Nazir, Majd Sakr, Lee Stott, and Carmen Taglienti. 2020. Cloud Computing Curriculum: Developing Exemplar Modules for General Course Inclusion. In *Proceedings of the Working Group Reports on Innovation and Technology in Computer Science Education (ITiCSE-WGR '20)*. Association for Computing Machinery, New York, NY, USA, 151–172. DOI: https://doi.org/10.1145/3437800.3439206 DOI: https://doi.org/10.1145/3304221.3325536

Prepared by: Yogesh Simmhan (IISc Bangalore), Purushottam Kulkarni (IIT Bombay)

******

4. Human Computer Interaction (HCI) Micro Specialization

Goal: The aim of this specialization is to provide an understanding of the importance and role of user-centered design and associated techniques.

Learning outcomes of the specialization:

1. Understand the user-centered design process and associated techniques to use it for designing and developing computing based solutions
2. Develop an ability to start with an observable problem in real-life settings and to develop a technology-led solution for the same.
3. To develop soft skills such as empathy for societal problems, the ability to think from another perspective, learning to work in a group
4. To learn methods related to observation, interviewing, problem identification, ideation, prototyping, and evaluation.
5. To apply the design process in a hands-on scenario.

Base Course for this specialization (core course): Advanced Programming

Courses (Electives) Proposed for the Specialization:

Course Name (full / half)	Purpose/Goal	Topics
Introduction to HCI (full)	To provide an understanding of the importance and role of user-centered design and techniques like observation, interviewing, problem identification, ideation, prototyping, and evaluation.	Contextual Inquiry, Interviews, Surveys, Focus Groups, sketching, low-fidelity prototyping, usability evaluation
HCI semester project (half)	Give an opportunity to apply the concepts learnt in the theory course in a hands-on project	Students will work on a project from real-life-like settings should be chosen and a group of students should be guided through the design process
Interactive Systems (half)	To develop a new interaction technique to solve a specific problem in HCI	Using any of the modern-day technologies, like, Virtual Reality, Augmented Reality, Speech technologies, Mobile programming, Drones, Haptic systems, Eye-trackers,

		etc. students will develop and test new interaction modalities.

Notes:

1. If there is faculty to offer further courses, then one/two other half courses can be added to this specialization – e.g. Development, Inclusive Design, Learning Technologies.

Prepared by: Pushpendra Singh (IIIT-Delhi), Anirudha Joshi (IIT Bombay)

******

5. Advanced Mobile Communications Micro Specialization

Goal: The aim this specialization is to provide an understanding of the advanced mobile communications technologies, in particular the 5G technology. As India and the world migrate to 5G, with India wishing to take a leading role in 5G technologies, trained manpower in these technologies will be necessary.

Module 1: Evolution from 1G to 5G (11 hours)

Analog voice systems in 1G; digital radio systems in 2G, voice and messaging services, TDMA based GSM, CDMA, 2.5G (GPRS), 2.75G (EDGE); IMT2000: 3G UMTS, W-CDMA, HSPA,
HSPA+, 3G services and data rates; IMT Advanced: 4G, LTE, VoLTE, OFDM, MIMO, LTE Advanced Pro (3GPP Release 13+); IMT2020: 5G, enhancements in comparison to IMT Advanced.

Module 2: Basics of 5G (11 hours)

5G potential and applications; Usage scenarios: enhanced mobile broadband (eMBB), ultra reliable low latency communications (URLLC), massive machine type communications (MMTC), D2D communications, V2X communications; Spectrum for 5G, spectrum access/sharing; millimeter Wave communication, channels and signals/waveforms in 5G, carrier aggregation, small cells, dual connectivity.

Module 3: 5G Network (14 hours)

New Radio (NR), Standalone and non-standalone mode; non-orthogonal multiple access (NOMA); massive MIMO, beam formation, FAPI: PHY API Specification, flexible frame structure, Service Data Adaptation Protocol (SDAP); centralized RAN, open RAN; multi-access edge computing (MEC); software defined networking (SDN), network function virtualization (NFV); network slicing; restful API for service-based interface; private networks.

Module 4: Current state and Challenges ahead (6 hours)

5G penetration in developed countries; deployment challenges in low-middle income countries, stronger backhaul requirements, dynamic spectrum access and usage of unlicensed spectrum, contrasting radio resource requirements; large cell usage: LMLC; possible solutions for connectivity in rural areas (BharatNet, TVWS, Long-range WiFi, FSO); non-terrestrial fronthaul/backhaul solutions: LEOs, HAP/UAV.

Suggested Reference Books

- 4G, LTE-Advanced Pro and The Road to 5G by Erik Dahlman
- 5G NR: Architecture, Technology, Implementation, and Operation of 3GPP New Radio
- Standards Hardcover – 1 June 2019 by Sassan Ahmadi Dr. (Author)

Appendix – II

A Guide to Induction Program

Appendix – II: A Guide to Induction Program

Introduction

In its 49th meeting, held on 14th March 2017, AICTE approved a package of measures for further improving the quality of technical education in the country. This 3-week mandatory Student Induction Program (SIP) based on Universal Human Values (UHV) is one of these key measures.

The SIP is intended to prepare newly admitted undergraduate students for the new stage in their life by facilitating a smooth transition from their home and school environment into the college and university environment.

The present form of the Student Induction Program (SIP) has taken inspiration from and gratefully acknowledges the many efforts in this direction. In particular the Foundation Program at IIT Gandhinagar[1] (July 2011) and the course in Universal Human Values and Professional Ethics[2] (IIIT Hyderabad, 2005; AKTU Lucknow, 2009 and PTU Jalandhar, 2011; overall about 35 universities); and also, the mentorship, internship and apprenticeship programs[3] of several institutions. The SIP amalgamates all the three into an integrated whole, which leads to its high effectiveness in terms of building a healthy lifestyle, creativity, bonding and character. It develops sensitivity towards self and one's relationships, builds awareness about others and society beyond the individual, and also in bonding with their own batch-mates and senior students as well as faculty members.

The purpose of this document along with accompanying details are to help institutions / colleges in understanding the spirit of the Induction Program and implementing it.

It is in line with the thoughts expressed in the NEP 2020:

"*Education is fundamental for achieving* ***full human potential****, developing an* ***equitable and just society****, and promoting* ***National development***".

"The purpose of the education system is to develop good human beings capable of rational thought and action, possessing compassion and empathy, courage and resilience, scientific temper and creative imagination, with sound ethical moorings and values".

[1] IIT Gandhinagar places great emphasis on not only educating successful engineers of the future, but also creating well-rounded personalities, who contribute to society, are respectful of and can adapt to their surroundings, and prove themselves to be great thinkers and problem solvers in all avenues of life. In 2011, in line with this vision, It took the bold step to introduce a five week Foundation Program for incoming 1st year UG students. It involved activities such as games, art, etc.; also science and other creative workshops as well as lectures by eminent resource persons. To enable undivided attention on this, normal classes were scheduled only after this program was over.

[2] The foundation course was started in 2005 at IIIT Hyderabad. In 2009, UP Technical University (now AKTU) introduced it in all academic programs across their 550 colleges. From there on, it has been included in the curriculum of many universities, particularly in technical universities, in quite a natural manner, filling a long-felt need. After AKTU, it was IKG-Punjab Technical University in 2011, then Royal University of Bhutan in 2012 and so on. By 2020, more than 40 universities in India and both universities of Bhutan have been offering this foundation course. Since 2017, it has been a compulsory credit course in AICTE's model curriculum for all UG courses. Faculty from all departments are involved in conducting the course. The content is universal, rational, verifiable and leading to harmony. The mode is a self-exploration (and not sermonising or lecturing). Faculty are to be prepared beforehand. The results have been quite encouraging.

[3] Many institutes setup mentor-mentee network under which 1st year students are divided into small groups, each assigned to a senior student as a Student Buddy, and to a faculty member as a Faculty Mentor. Thus, a new student has their guidance through regular interactions. They can discuss their aims and aspirations as well as concerns whether social, psychological, financial, academic, or otherwise.

"It aims at producing engaged, productive, and contributing citizens for building an equitable, inclusive, and plural society as envisaged by our Constitution".

"Education must build character, enable learners to be ethical, rational, compassionate, and caring, while at the same time prepare them for gainful, fulfilling employment".

"The curriculum must include basic arts, crafts, humanities, games, sports and fitness, languages, literature, culture, and values, in addition to science and mathematics, to develop all aspects and capabilities of learners; and make education more well-rounded, useful, and fulfilling to the learner".

So, when new students join an institution, they are to be welcomed and oriented to the institute, its vision, people, purpose, culture and values, policies, programs, rules and regulations etc. through a well-planned 3-week interaction before regular classes start.

Education aims at developing the students to their full potential, so that they are able to participate meaningfully not only in their profession, but also in their family, society and their natural environment. That requires the development of their values as well as skills.

Engineering colleges were established to train graduates in their respective branch/ department of study, be ready for the job market, but also have a holistic outlook towards life and have a desire and competence to work for national needs and beyond. The graduating student must have the knowledge and skills in the area of his study. However, s(he) must also have a broad understanding of society and relationships. Besides the above, several meta-skills and underlying values are needed. Character needs to be nurtured as an essential quality by which s(he) would understand and fulfil his/her responsibility as an engineer, a family member, a citizen etc.

The same applies to all other branches of study – be it professional, vocational or any other area of academic. The graduating student must be a good human being and have the skills in their area of study.

Each family, institution, region, community etc. have evolved their way of life, their cultures over a period of time. The new students are going from one culture to another. Today, a major issue is that one culture tends to be opposed to other cultures. This is because their basic assumptions, and therefore thoughts, are different. Even though there are commonalities at the core value level, the conflict is at the level of expression and details.

With this situation, it is imperative to

- Articulate the essence or core aspects of human culture and civilization, i.e. understand universal human values like trust and respect, love and compassion
- Appreciate the various expressions, different approaches taken in different regions

Our effort is in the context of the whole humanity. However, when it comes to exemplifying these essential concepts, we will have to take to local or national expressions.

In SIP, we want to provide an exposure to essence in the context of the whole humanity first. Then we can take a representative cross-section of all cultures as expressions of this essence.

A yardstick to evaluate these various options is provided to guide the student towards a humanistic culture founded on the truth and universal human values like love and compassion.

For example: We want to live with fulfilment as a society. This part is common, universal.
To exemplify this, we may expose students to traditional Indian culture and philosophy as well as contemporary western culture and thought.

The intent is:

- Connecting the basic principles through specific examples
- To see and appreciate various cultures, to see the commonality amongst them, in the light of clarity about human culture and civilisation.
- To evaluate any specific example, system or culture, with a view to fill the gaps, rather than to criticise or reject it. Further, we can also be mutually enriching for other cultures.

Student Induction Program (SIP)

With this background, the SIP has been formulated with specific goals to help students to:

- Become familiar with the ethos and culture of the institution (based on institutional culture and practices)
- Set a healthy daily routine, create bonding in batch as well as between faculty members and students
- Get an exposure to a holistic vision of life, develop awareness, sensitivity and understanding of the
  Self---family---Society---Nation---International---Entire Nature
- Facilitate them in creating new bonds with peers and seniors who accompany them through their college life and beyond
- Overcome weaknesses in some essential professional skills – only for those who need it (e.g. Mathematics, Language proficiency modules)

The SIP consists of different activities which includes meeting new students, socializing with teachers and other people in the university. Secondly associating with the Local area or city, knowing different departments, associating with the department heads, local stores and necessary shops for the survival at new place. Basically, getting information about the rules and regulations of the university which includes do's and don'ts. Other activities which may involve students in several creative, cultural and co- curricular activities through which they can explore themselves and get idea about their intrinsic desires and interests which may help them in the long run. In order to make it worth, at the initial level of joining of student various seminars, lectures by eminent personalities, sessions by the appointed mentor for the student is being done to make them more familiar with the university environment. It has been seen that student after schooling when moves towards further studies for either under graduation or post-graduation has got so many confusions and false knowledge about the college and the curriculum. They should know the basic idea about the fruits and prospects of the particular course and the university or institute in which they are entering. To have faith about their choices and to know that after completion, they will be well equipped with the values and skills which may aid to their future goals and let them work for their personal motives, society and the Nation's development.

The various modules or core areas recommended for the 3-week SIP are:

SIP Module 1: Universal Human Values I (UHV I) 22 hours

The purpose is to help develop a holistic perspective about life. A self-reflective methodology of teaching is adopted. It opens the space for the student to explore his/her role (value) in all aspects of living – as an individual, as a member of a family, as a part of the society and as a unit in nature. Through this process of self-exploration, students are able to discover the values intrinsic in them. The session-wise topics are given below:

Session No.	Topic Title	Aspirations and Issues	Basic Realities (underlying harmony)
1	Welcome and Introductions	Getting to know each other	Self-exploration
2 and 3	Aspirations and Concerns	Individual academic, career… Expectations of family, peers, society, nation… Fixing one's goals	Basic human aspirations Need for a holistic perspective Role of UHV
4 and 5	Self-Management	Self-confidence, peer pressure, time management, anger, stress… Personality development, self-improvement…	Harmony in the human being
6 and 7	Health	Health issues, healthy diet, healthy lifestyle Hostel life	Harmony of the Self and Body Mental and physical health
8, 9, 10 and 11	Relationships	Home sickness, gratitude towards parents, teachers and others Ragging and interaction Competition and cooperation Peer pressure	Harmony in relationship Feelings of trust, respect… gratitude, glory, love
12	Society	Participation in society	Harmony in the society
13	Natural Environment	Participation in nature	Harmony in nature/existence
14	Sum Up	Review role of education Need for a holistic perspective	Information about UHV-II course, mentor and buddy
15	Self-evaluation and Closure	Sharing and feedback	

SIP Module 2: Physical Health and Related Activities 51 hours

This module is intended to help understand the basic principles to remain healthy and fit and practice them through a healthy routine which includes exercise, games etc.

SIP Module 3: Familiarization of Department/ Branch and Innovation 06 hours

This module is for introducing and relating the student to the institution/department/branch; how it plays a role in the development of the society, the state, region, nation and the world at large and how students can participate in it.

SIP Module 4: Visit to a Local Area **10 hours**

To relate to the social environment of the educational institution as well as the area in which it is situated through interaction with the people, place, history, politics…

SIP Module 5: Lectures by Eminent People **06 hours**

Listening to the life and times of eminent people from various fields like academics, industry etc. about careers, art, self-management and so on enriches the student's perspective and provides a holistic learning experience.

SIP Module 6: Proficiency Modules **06 hours**

This module is to help fill the gaps in basic competency required for further inputs to be absorbed. It includes effort to make student proficient in interpersonal communication and expression as well as awareness about linguistic and thereafter NLP.

SIP Module 7: Literature / Literary Activities **30 hours**

Through the exposure of local, national and international literature, this module is aimed at helping the student learn about traditional as well as contemporary values and thought.

SIP Module 8: Creative Practices **49 hours**

This module is to help develop the clarity of humanistic culture and its creative, joyful expression through practice of art forms like dance, drama, music, painting, pottery, sculpture etc.

SIP Module 9: Extra Curricular Activities **06 hours**

This is a category under which things that are not placed in any of the above may be placed. Some clubs and hobby group may be made for each of the above categories, so that students may pursue them even after SIP.

The recommended hours to be allocated are given above. Depending on the available faculty, staff, infrastructure, playgrounds, class timings, hostellers and day scholars etc., the timetable for these activities may be drawn up. Of course, colleges may conduct an inaugural function at the beginning of the SIP; and they may also conduct a celebratory closing ceremony at the end of the SIP.

In particular, during the lockdown phase, appropriate care may be taken and some or all activities may be planned in distance-learning or on-line mode.

Sample 3-week Activity List

Week 1	Inaugural Function Regular SIP Activities (See Hours Plan)
Week 2	Regular SIP Activities (See Hours Plan)

Week 3	Regular SIP Activities (See Hours Plan) Valedictory and Closing Ceremony (Celebration)

Implementation

Every institution/college is expected to conduct the 3-week SIP under the guidance of the Director/Principal or Dean Students or a senior faculty member. For this, the institution is expected to make an SIP Cell / team, which will be responsible for planning, and then implementation of the SIP.

A UHV Cell is expected to be set up at each college and university. At the college, it will be managed by the UHV Convener / Coordinator under the chairpersonship of the director/principal. Faculty members and some students will be the members. They will coordinate the UHV activities like UHV-I during SIP, UHV-II, the faculty mentoring program and student buddy program throughout the student's association with the institute/college. The UHV Cell will work to incorporate human values in every aspect of education at the institute/college. Preparing UHV Faculty (Mentors) is one of its important activities.

Follow up

The SIP is only the beginning of the interaction with newly joined students.

An important part of the SIP is to associate one faculty mentor to every small groups of about 20 students; and also associate one senior student buddy to an even smaller groups of about 5 students for the guidance required for holistic development of the newly joined student throughout his/her time in the institution/college.

These activities are to be continued in the ongoing academic program along with other cultural activities through various student clubs which are largely be managed by students with the help of one or more faculty mentors. One of the main responsibilities of the faculty mentors would be helping the clubs to review their activities in alignment with human values.

Assessing the Implementation and Impact

The institution / college is expected to take feedback and prepare appropriate reports for assessing the impact and for further improvement of SIP. The basic feedback forms are included with the SIP Teaching Materials.

The SIP and its further follow up is expected to positively impact common graduate attributes like:

- Holistic vision of life
- Socially responsible behaviour
- Environmentally responsible work
- Ethical human conduct

Having Competence and Capabilities for Maintaining Health and Hygiene
Appreciation and aspiration for excellence (merit) and gratitude for all

AICTE will conduct periodic assessment to ascertain the implementation efforts and impact of the SIP and related activities.

Faculty Development

To ensure the implementation of SIP, and in particular to prepare the faculty, the National Coordination Committee for Student Induction (NCC-IP) has been formed. It offers various faculty development programs (FDPs) with the support from AICTE HQ and Regional Offices.

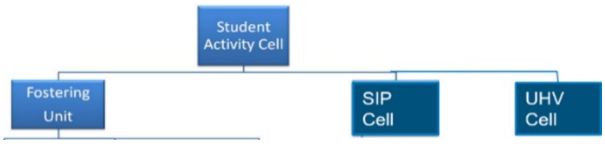
UHV Faculty (Mentors): Every institution is expected to prepare UHV Faculty in the ratio of 1:20 (1 faculty per 20 newly admitted students). Faculty from every teaching department are to be prepared. The basic preparation is participation in an 8-day FDP-SI (UHV).

Faculty for other Modules: Institutions/colleges generally have faculty, coaches, student clubs, alumni for these areas. FDP and comprehensive material will also be made available.

Student Activity Cell (SAC) – SIP Cell, UHV Cell and Fostering Unit

Student Activity Cell will have three cells or units:

- Fostering Unit – for coordinating various student clubs and activities in alignment with human values and IKS through various student clubs
- SIP Cell – for coordinating the annual SIP
- UHV Cell – for coordinating regular UHV activities, including UHV-I during SIP and UHV-II during future semesters, faculty mentoring and student buddy programs etc.



Each cell / unit will have some axis. E.g. the Fostering Unit will have 3 axis:

- UHV Axis – for UHV inputs and activities after the SIP
- Health Axis – for health oriented inputs and activities after SIP
- Career Axis – for career related inputs

Each axis will have one or more dimensions. E.g. the UHV Axis will have two dimensions:

- UHV Dimension
- Social Work Dimension

- Details of the clubs will be based on local conditions.

- Director or Principal or Dean of Student affairs will be the Chairman of Student Activity Cell

- SIP Cell (or Induction Unit) will be managed by faculty members with the help of student volunteers. 5 to 7 faculty members will be the members. The SIP Cell will be responsible for planning, organization, coordination and reporting of the annual Student Induction Program with the help of other faculty members and student volunteers

- UHV Cell will be managed by the UHV Convener / Coordinator under the chairpersonship of the director/principal. Faculty members and some students will be the members. They will coordinate the UHV activities like UHV-I during SIP, UHV-II 3rd/4th semester, faculty mentoring program and student buddy program throughout the student's association with the institute/college. UHV Cell will work to incorporate human values in every aspect of education at the institute/college. Preparing UHV Faculty (Mentors) is one of its activities

- Fostering unit will largely be managed by students with the help of one fostering unit faculty mentor. Student will be coordinators for axis, dimensions and clubs. Fostering unit will take support from induction unit as and when required. It will be responsible for coordinating various student clubs and activities in alignment with human values and Indian Knowledge System

SIP Teaching Material and More Details

The SIP Handbook as well as detailed guides and material for each of the modules is available on the AICTE website (http://www.fdp-si.aicte-india.org/download.php).

Details and Reference Documents:

- G012 SIP Handbook v2
- Teaching Material for UHV-I v2.1
- Teaching Material for SIP modules 2 to 9 v1
- G008 Facilitator (Mentor) Manual Version 2.1
- G911 UHV Cell, Nodal and Resource Centres
- G009 RP Development Process v2

#Note: For CSE UG Students only

The Department of Telecommunications, Ministry of Communication, Government of India is going to auction 5G spectrum shortly. The adoption of 5G will accelerate employment generation in telecom and technology industry. The 5G Technology will penetrate the entire telecom ecosystem of hardware, software and services that are critical for implementation of other futuristic technologies like Internet of Thing (loT), Machine-to-Machine **(M2M}** communication, edge computing etc. Innovative applications in various sectors like agriculture, transportation, power etc.

will use and requires knowledge of inherent features of 5G. There will be huge requirement of market ready talent pool in 5G technology.

Considering the need for specialized courses and modules on SG Technology, National Telecommunication Institute for Policy & Research, Innovation & Training (NTIPRIT)-Department of Telecommunication, after due consultation with academia and industry, sent a proposal to AICTE vide No. 1-3/2020-NTI.TS-SD dated 09.03.2021 to include the following:

- A full Semester course on "Advanced Mobile Communications" for UG
- A 14-hour 5G awareness Program for UG Students;

5G Awareness Programme for UG students (14 hours)

Course Title: Introduction to 5G

Topics to be covered

1. IMT2020 enhancements in comparison to IMT Advanced
2. 5G potential and applications
3. Usage scenarios: eMBB, URLLC, MMTC
4. Spectrum for 5G and spectrum sharing
5. Millimeter wave communication and small cells
6. New Radio: SA and NSA mode
7. Massive MIMO and beam forming
8. Multi-access edge computing
9. Software defined networks
10. Network slicing
11. Current state of deployment
12. Large cell scenarios: LMLC

*****

Appendix 3: Recommendations for Using Online Content in Courses at Colleges and Universities

Background

A lot of online high quality content is available today either free or at a low cost. Besides the government supported NPTEL, we have companies who aggregate courses from several universities (and even private commercial organizations) and offer them to students all over the world. There are other companies who offer courses designed and developed by them. Further, a lot of companies have online content available to students.

On the other hand, most of the Computer Science (and related) departments face serious shortage of faculty, particularly in areas where there is a significant demand in the industry.

So, on one hand, we have quality content available for free or low cost, and on the other hand, we don't have faculty to teach such courses. The natural solution is to find ways to use online content for the courses in the curriculum (with credits). This way, either a knowledgeable faculty can "teach" a much larger class, or a faculty member with inadequate background in the topic can still "teach" the course better than what s/he would have done without the support of such online content.

Issues

The online content has been around for several years. The regulatory bodies have also been encouraging use of such content (particularly, NPTEL). And yet, the online content has not been integrated with the curriculum in most colleges. The pandemic has allowed people to take a fresh look at online content and the mental barriers to using such content in the curriculum have been breached. At this time, it is felt that a lot of colleges would want to use this content. However, there are two primary academic issues that need to be addressed (besides logistics, financial, and HR issues). These are:

1. How do we decide what material to use? This has two sub-issues. One, what content would be equivalent to the content that is mentioned in our curriculum (course mapping). Note this is a challenge because there may not be a single module which covers all parts of the curriculum. So we may need to select more than one module. Two, given the plethora of content, which content is of reasonable quality.
2. How would a college do evaluation of students in order to assign marks/grades?

Modes of Learning

There are several ways of using the online content in the curriculum.

The simplest mode (Mode 1) is to use online content as additional reference material. In this mode, the normal teaching is anyway being done and the faculty is referring to online content in the same way he/she would refer to a book. This mode requires no change or suggestions and hence is outside the scope of this document.

The next mode (Mode 2) is the flipped classroom model where the students go through the online content (including writing programs, small quizzes, etc.), and the local faculty takes discussion sessions and does all the evaluation. This mode still requires a knowledgeable faculty member to be the instructor. However, given that we can now reduce the contact hours for the students and faculty, the same faculty can handle a much larger class (or multiple sections in case there is an upper limit on the size of the class).

The next mode (Mode 3) is that the students are studying only through the online mode and there is very little interaction with the faculty at the college. May be there can be some sessions once in a while, but mostly, the role of a local faculty is only to handle evaluation (exams, assignments, projects, etc.). This partially addresses the issue of lack of faculty in certain areas since the expertise required for handling evaluation is arguably lesser than the expertise required to teach the course.

The last mode (Mode 4) is where the online provider does everything, including evaluations. We don't need any faculty member at the college to offer this course. We only need to decide how to translate the evaluation done by an outsider to an equivalent grade/marks on the college transcript.

The table below illustrate the four models in brief.

S.No.	Model	Local Part	Online Part	Guidelines
1	Reference Mode	All lectures, home works, evaluations	Additional reference material	No additional recommendations. This requires no additional details.
2	Flipped Mode	Discussion sessions, evaluations	Lectures, Home works, Quizzes, Project options,	Requires local faculty with some knowledge in the subject area Large sections/class sizes can be handled
3	Screen Mode	Evaluations	All lectures, Home works, Projects	Place reasonable limit on the number of credits that can be earned in this mode Use in cases where there is faculty shortage in critical areas
4	Fully online	--	All lectures, evaluations, Projects, Tests	Place reasonable limit on the number of credits that can be earned in this mode. Advised not to repeat this model for the same course more than two times. Local faculty also to register and go through the course. Official score/grade to be suitably added to the student transcript

Course Mapping

Each course in the curriculum has course outcomes and the content defined by the university. Typically, we may have some course outcomes which are important ones, and may be some outcome which is desirable or optional. Similarly, the curriculum may also define the rough duration of each topic. Sometimes the curriculum will also include the kind of projects or assignments that the student shall be asked to do in the course.

The key to course mapping is to realize that any two individuals defining a course will have some differences and we must be flexible to accept differences to some extent. The committee trying to do course mapping should have an understanding of what are the important components (in all three: outcomes, topics, and projects) and should ensure that the online content meets all the important requirements. But looking for identical course would be usually futile.

Also, what is noticed is that typical online courses are often available in smaller modules than a typical 4-credit course in our curriculum. And therefore, one may need to consider more than one online courses together to be equivalent to one course in the college. It may also be noted that just like some small aspects of the course may not be present in the modules chosen, there may be some aspects of the online modules which were not part of the college course. This amount of flexibility should be acceptable to the college. Typically, if the online content covers 80% of the college course, it may be accepted.

While autonomous colleges and universities do have this flexibility, the affiliated colleges may not have the flexibility of not teaching even 20% of the content. In case of affiliated colleges, one will have to either be stricter in course mapping, or find a way to cover the gaps through a local faculty or a visiting faculty. It is assumed that it is easier to find a visiting faculty for a small part of the course and hence it is still a useful mode.

It may also be noted here that for Core courses, the overlap needs to be significantly higher while for the elective courses, the overlap could be relaxed somewhat. It is because the core courses typically are pre-requisites for other courses. Also, core courses have been defined to be such because it is assumed that that content is more important for the graduates than what is taught in electives.

However, course mapping is still not an easy thing to do. It requires an understanding of important versus less important components, and quality of content among the plethora of options available. It may require someone to go through the content patiently.

And hence it is recommended that for standard courses recommended in the AICTE model curriculum, a course mapping may be suggested for the benefit of colleges and universities. A few example course mappings are, therefore, attached with this report.

Evaluation

In Modes 1, 2, and 3, the complete evaluation is local, and hence there is no issue. In Mode 4, we need to consider an external evaluation and use that internally. This is a challenge. There is a difference in how to handle this in a university versus an affiliating college. In a university, a simple way of handling this would be to assign a Pass/Fail grade to the student. In case of an affiliating college, where only the internal marks need to be forwarded to the university, one could consider the external evaluation since it impacts only 30% of the marks, and the university will anyway have its own exam of 70% marks.

There are other models like normalizing the external evaluation to the college's internal policy or average distribution of marks in other courses.

It is assumed that in Mode 4, there is a formal way of communicating performance of the student by the provider of the online courses since colleges and universities will not accept screen shots, emails, etc.

Other Issues

Financial: If there is a fee to be paid for online content, the college should have a policy on that. Typically, if the savings due to reduced faculty requirements are significant, then the fees for the online provider may be reimbursed by the college.

Teaching Load: Faculty member supporting the course whether by taking a few discussion sessions (in Mode 2) or by evaluating the students (in Mode 3) is still putting in substantial effort in managing the course, and an appropriate credit should be given to the faculty member when his/her teaching load is computed.

Training the teacher: When a course is being done in Mode 3, the local faculty member should also be expected to register for the course and go through the course (with load being appropriately counted). After a faculty member has gone through the course in two academic sessions, s/he would be well prepared to teach the course in the class in a much better way. Even if the course is being offered in Mode 4, there is no harm in asking a faculty member to register for the course and go through it. Some responsible person in the college would know the level of the course and what exactly students have done, and again, after two such sessions, the faculty member would be well prepared to teach the course. Hence this mode will also lead to better training of the teachers.

Limits on Credits: The committee believes that there should be a limit on the number of credits students can earn through online courses. In case of Mode 4, where even the evaluation is done by the online course provider, the proposed limit is 8 credits only. It is felt by the committee that the evaluation by online providers is still not fully trustworthy. As the technology for online exams or the processes for evaluation by online providers improve over a period of time, this limit may be increased. In case of Mode 3 where the content is delivered online but evaluation is local, the limit can be high. For Mode 3 and Mode 4 combined, the limit can be what the regulatory bodies like UGC have announced for online courses, which is currently 40% of the total credits. In Modes 1 and 2 where the online content is really the reference material, there is no need for any limit.

Another constraint the committee would want the colleges to consider is that in a sequence of courses in one stream of Computer Science, at least one course should be in class. For example, if we consider the sequence of systems courses – Operating Systems, Databases, Networks, Architecture, at least one course should be in class. This is to ensure that if there were some gaps in online courses, the faculty in the face to face class can try to cover that to some extent.

Faculty Incentive: There is a need to provide some incentive to faculty members who would manage the course that is being taught in the online mode. If a course is being taught in Mode 2, the load on the faculty is only marginally less than the load of teaching an in person course. So the full teaching load should be considered for the faculty. In Mode 3, the load is much less, and in Mode 4, the load is only that we are asking the faculty to also go through the course along with the students. In these two modes, the college may consider this as reduced load. However, their learning the course may be treated as equivalent to having done a Faculty Development Program when it comes to their appraisal and promotions.

Consideration in NBA Accreditation: One of the prime reason why online courses haven't become popular with colleges is that they must recruit faculty with a certain faculty to student ratio for accreditation and ranking. And once they have recruited faculty, one would always want the faculty to teach and not keep them under-loaded. If one can consider online courses as equivalent to faculty strength while deciding faculty-to-student ratio, then colleges would be attracted to online courses. A typical faculty member teaches about 100 students in a semester (across 2-3 courses). If 100 students do a course in Mode 4, we may consider this as equivalent to having one additional full time equivalent (FTE) faculty member on the rolls of the college for that semester. Similarly, if 200 students do a course in Mode 3, we may consider this as equivalent to having one additional FTE faculty member for that semester.

Online offering of AICTE Model Curriculum Courses by Experts

It is known that a significant number of colleges do not have adequately qualified faculty to teach the courses in the model curriculum properly. This proposal tries to alleviate this challenge by utilizing our newly acquired comfort level with online education. Essentially, the suggestion is that AICTE incentivizes top faculty to offer the AICTE model courses online, and publishes a calendar of such courses well in advance, so colleges/HEIs can use them for teaching the AICTE curriculum.

The following would be the salient features of the proposed scheme

1. All the identified core courses and later on some key electives of the AICTE curriculum may offered regularly (maybe every semester) with a predefined calendar enabling colleges to plan using them for their students.
2. For each course AICTE identifies through a process a set of "Distinguished National Technical Teachers" who are subject experts and who also have experience teaching the course at the undergraduate level.
3. These "Distinguished Teachers" are offered financial reward for offering one course online once every 2 to 3 years. The reward must be substantial to make it prestigious as well as attractive. Suitable funding is also made available to the instructor for engaging TAs for help sessions as well as laboratory exercises.
4. AICTE can also recognize industry professionals in the panel of distinguished faculty as they can make these courses more interesting and industry oriented. They could teach jointly with academics appointed as Distinguished Teachers and handle certain components of the course. The Distinguished Teachers will be encouraged to actively invite guest faculty from industry and/or professional associations (e.g. ACM) to give lectures in these courses to make them more interesting and industry oriented.
5. There is a registration procedure for colleges (not individual students) to register for making this course available to their students. They should also commit to assigning an instructor for coordinating classes including infrastructure, conducting assessments, help in grading etc.
6. Colleges may be asked to pay a reasonable amount as registration charges so that the scheme is self-sustaining.
7. This will be an AICTE scheme for teaching its model curriculum courses by experts. It can be managed by AICTE directly, or it can offer it to other institutions/consortium to manage them. Any platform (e.g. NPTEL) can be used to deliver these courses.

This approach differs from the currently available online courses from NPTEL and other platforms substantially.

1. Syllabus for each course is the approved AICTE syllabus
2. The courses would be available against a set calendar each semester making it possible for colleges to rely on them
3. It empowers the college rather than individual students to make use of the online course in a "mixed" mode and thus addresses the deficiency of expertise in specific areas in that college
4. With suitable incentive (both prestige and financial) top experts may be motivated to offer these courses.

Prepared by: Dheeraj Sanghi (JKLU), Kishore Kothapalli (IIITH), Ashalata Nayak (MAHE-MIT), Suchismita Roy (NIT Durgapur), RBV Subramanyam (NITW), Divya Bansal (PEC), M. Balakrishnan (IIT Delhi)

Appendix 4

Recommendations for Possible Exits for a B Tech CSE

Preamble

NEP2020 suggests that a student should have multiple exits. This note suggests a possibility for B Tech CSE students. It should be emphasized that it is a choice which a student may wish to take due to his/her financial/family/other situation and needs, and that it should not be considered as a failure option. We suggest two exits, and flexible re-entry options.

Certificate in Computer Science

A student should be able to get a certificate if he/she completes:

1. 50% of the credits for B Tech
2. 50% of CSE program core courses (some specific courses may be specified, so the leaving student has decent skills)
3. Institution may specify some CGPA requirements

BSc in Computer Science

A student should be able to get a BSc if he/she completes:

1. 75% of the credits for B Tech, and at least 3 years in the program
2. 100% of CSE core program courses
3. Institution may specify some CGPA requirements

With BSc, the student is eligible for entry into programs which take BSc as eligibility criteria.

Re-entry to complete the program

A student exiting with a certificate or BSc should be entitled to re-enrol in the program. It is suggested that all credits will be transferred, if the student enrols back within a limited period (suggested: 3 years) of exiting. In case a student enrols after that, then the transfer of credits should be examined by looking at the change in the curriculum from what the student did.

Completion Possibility in other Institutions

It will be desirable for HEIs to make it possible for a student to earn a certificate/BSc in one institution and complete the degree program in another. This will enhance the value of certificates and BSc and encourage competition among HEIs.

If these exit options are accepted for multiple B Tech programs, it is suggested that AICTE actively communicate these to the industry and other bodies, so they recognize these and accept them as bona-fide credentials for the purposes of recruitment and/or eligibility for admission to programs, appearing in competitive examinations, etc.

Prepared by: Prof. M. Balakrishnan (IIT Delhi), Prof. Anshul Kumar (IIT Delhi), Prof. Pankaj Jalote (IIIT-Delhi)

Appendix- 5:
Mathematics- III
(for Computer Science & Engineering students)

MATHEMATICS 3	**2L:0T:0P**	**2 CREDITS**

This will be a need based customized course for different branches with modules chosen from the following:

1. Partial Differential Equations (5c, 5d)
2. Numerical Methods (Modules 7a, 7b)
3. Transform Calculus ((Modules 8a, 8b)
4. Discrete Mathematics (Modules 9a, 9b, 9c, 9d, 9e, 9f)
5. Probability and Statistics (Modules 10a, 10b, 10c, 10d, 10e, 10f)

If needed, there can be an additional Mathematics course:

MATHEMATICS 4: (2-0-0-2)

All India Council for Technical Education
Mathematics Courses, Option 2 (for CSE etc.)

Paper 1: Calculus and Linear Algebra (3-1-0-4)

Module 1: Calculus: (6 hours)
Evolutes and involutes; Evaluation of definite and improper integrals; Beta and Gamma functions and their properties; Applications of definite integrals to evaluate surface areas and volumes of revolutions.

Module 2: Calculus: (6 hours)
Rolle's theorem, Mean value theorems, Taylor's and Maclaurin theorems with remainders; Indeterminate forms and L'Hospital's rule; Maxima and minima.

Module 3: Matrices (in case vector spaces is to be taught) (8 hours)
Matrices, vectors: addition and scalar multiplication, matrix multiplication; Linear systems of equations, linear Independence, rank of a matrix, determinants, Cramer's Rule, inverse of a matrix, Gauss elimination and Gauss-Jordan elimination.

Module 4: Vector spaces (Prerequisite 4b) (10 hours)
Vector Space, linear dependence of vectors, basis, dimension; Linear transformations (maps), range and kernel of a linear map, rank and nullity, Inverse of a linear transformation, rank-nullity theorem, composition of linear maps, Matrix associated with a linear map.

Module 5: Vector spaces (Prerequisite 4b-c) (10 hours)
Eigenvalues, eigenvectors, symmetric, skew-symmetric, and orthogonal Matrices, eigenbases. Diagonalization; Inner product spaces, Gram-Schmidt orthogonalization.

Textbooks/References:

1. G.B. Thomas and R.L. Finney, Calculus and Analytic geometry, 9th Edition, Pearson, Reprint, 2002.
2. Erwin Kreyszig, Advanced Engineering Mathematics, 9th Edition, John Wiley & Sons, 2006.
3. D. Poole, Linear Algebra: A Modern Introduction, 2nd Edition, Brooks/Cole, 2005.
4. Veerarajan T., Engineering Mathematics for first year, Tata McGraw-Hill, New Delhi, 2008.
5. Ramana B.V., Higher Engineering Mathematics, Tata McGraw Hill New Delhi, 11th Reprint, 2010.
6. N.P. Bali and Manish Goyal, A text book of Engineering Mathematics, Laxmi Publications, Reprint, 2010.
7. B.S. Grewal, Higher Engineering Mathematics, Khanna Publishers, 35th Edition, 2000.
8. V. Krishnamurthy, V.P. Mainra and J.L. Arora, An introduction to Linear Algebra, Affiliated East–West press, Reprint 2005.

Paper 2: Probability and Statistics (3-1-0-4)

Module 1: **Basic Probability:** (12 hours)
Probability spaces, conditional probability, independence; Discrete random variables, Independent random variables, the multinomial distribution, Poisson approximation to the binomial distribution, infinite sequences of Bernoulli trials, sums of independent random variables; Expectation of Discrete Random Variables, Moments, Variance of a sum, Correlation coefficient, Chebyshev's Inequality.

Module 2: **Continuous Probability Distributions:** (4 hours)
Continuous random varibales and their properties, distribution functions and densities, normal, exponential and gamma densities.

Module 3: **Bivariate Distributions:** (4 hours)
Bivariate distributions and their properties, distribution of sums and quotients, conditional densities, Bayes' rule.

Module 4: Basic Statistics: (8 hours)
Measures of Central tendency: Moments, skewness and Kurtosis - Probability distributions: Binomial, Poisson and Normal - evaluation of statistical parameters for these three distributions, Correlation and regression – Rank correlation

Module 5: Applied Statistics: (8 hours)
Curve fitting by the method of least squares- fitting of straight lines, second degree parabolas and more general curves. Test of significance: Large sample test for single proportion, difference of proportions, single mean, difference of means, and difference of standard deviations.

Module 6: Small samples: (4 hours)
Test for single mean, difference of means and correlation coefficients, test for ratio of variances - Chi-square test for goodness of fit and independence of attributes.

Textbooks/References:

1. Erwin Kreyszig, Advanced Engineering Mathematics, 9th Edition, John Wiley & Sons, 2006.
2. P. G. Hoel, S. C. Port and C. J. Stone, Introduction to Probability Theory, Universal Book Stall, 2003 (Reprint).
3. S. Ross, A First Course in Probability, 6th Ed., Pearson Education India, 2002.
4. W. Feller, An Introduction to Probability Theory and its Applications, Vol. 1, 3rd Ed., Wiley, 1968.
5. N.P. Bali and Manish Goyal, A text book of Engineering Mathematics, Laxmi Publications, Reprint, 2010.
6. B.S. Grewal, Higher Engineering Mathematics, Khanna Publishers, 35th Edition, 2000.
7. Veerarajan T., Engineering Mathematics (for semester III), Tata McGraw-Hill, New Delhi, 2010.

Paper 3: Calculus and Ordinary Differential Equations (3-0-0-3)

Module 1: Sequences and series: (Prerequisite 2b) (8 hours)
Convergence of sequence and series, tests for convergence, power series, Taylor's series. Series for exponential, trigonometric and logarithmic functions.

Module 2: Multivariable Calculus (Differentiation) (Prerequisite 2b) (8 hours)
Limit, continuity and partial derivatives, directional derivatives, total derivative; Tangent plane and normal line; Maxima, minima and saddle points; Method of Lagrange multipliers; Gradient, curl and divergence.

Module 3: Multivariable Calculus (Integration) (Prerequisite 3a) (10 hours)
Multiple Integration: double and triple integrals (Cartesian and polar), change of order of integration in double integrals, Change of variables (Cartesian to polar). Theorems of Green, Gauss and Stokes, orthogonal curvilinear coordinates, Simple applications involving cubes, sphere and rectangular parallelepipeds.

Module 4: First order ordinary differential equations(6 hours)
Exact, linear and Bernoulli's equations, Euler's equations, Equations not of first degree: equations solvable for p, equations solvable for y, equations solvable for x and Clairaut's type.

Module 5: Ordinary differential equations of higher orders (8 hours)
Second order linear differential equations with variable coefficients, method of variation of parameters, Cauchy-Euler equation; Power series solutions; Legendre polynomials, Bessel functions of the first kind and their properties.

Textbooks/References:

1. G.B. Thomas and R.L. Finney, Calculus and Analytic geometry, 9th Edition, Pearson, Reprint, 2002.
2. Veerarajan T., Engineering Mathematics for first year, Tata McGraw-Hill, New Delhi, 2008.
3. Ramana B.V., Higher Engineering Mathematics, Tata McGraw Hill New Delhi, 11th Reprint, 2010.
4. N.P. Bali and Manish Goyal, A text book of Engineering Mathematics, Laxmi Publications, Reprint, 2010.
5. B.S. Grewal, Higher Engineering Mathematics, Khanna Publishers, 35th Edition, 2000.
6. Erwin Kreyszig, Advanced Engineering Mathematics, 9th Edition, John Wiley & Sons, 2006.
7. W.E. Boyce and R. C. DiPrima, Elementary Differential Equations and Boundary Value Problems, 9th Edition, Wiley India, 2009.
8. S.L. Ross, Differential Equations, 3rd Ed., Wiley India, 1984.
9. E.A. Coddington, An Introduction to Ordinary Differential Equations, Prentice Hall India, 1995.
10. E.L. Ince, Ordinary Differential Equations, Dover Publications, 1958.
11. G.F. Simmons and S.G. Krantz, Differential Equations, Tata McGraw Hill, 2007.

All India Council for Technical Education
Different Modules for Mathematics Courses

1. Algebra and Trigonometry (Optional)

Module 1a: Trigonometry: Hyperbolic and circular functions, logarithms of complex number, resolving real and imaginary parts of a complex quantity, De Moivre's Theorem.
Module 1b: Theory of equations: Relation between roots and coefficients, reciprocal equations, transformation of equations and diminishing the roots.

2. *Calculus (Single Variable)*

Module 2a: Calculus: (6 hours)
Evolutes and involutes; Evaluation of definite and improper integrals; Beta and Gamma functions and their properties; Applications of definite integrals to evaluate surface areas and volumes of revolutions.

Module 2b: Calculus: (6 hours)
Rolle's theorem, Mean value theorems, Taylor's and Maclaurin theorems with remainders; Indeterminate forms and L'Hospital's rule; Maxima and minima.

Module 2c:Sequences and series: (Prerequisite 2b) (10 hours)
Convergence of sequence and series, tests for convergence, power series, Taylor's series. Series for exponential, trigonometric and logarithmic functions; Fourier series: Half range sine and cosine series, Parseval's theorem.

Textbooks/References:

1. G.B. Thomas and R.L. Finney, Calculus and Analytic geometry, 9th Edition, Pearson, Reprint, 2002.
2. Veerarajan T., Engineering Mathematics for first year, Tata McGraw-Hill, New Delhi, 2008.
3. Ramana B.V., Higher Engineering Mathematics, Tata McGraw Hill New Delhi, 11th Reprint, 2010.
4. N.P. Bali and Manish Goyal, A text book of Engineering Mathematics, Laxmi Publications, Reprint, 2010.
5. B.S. Grewal, Higher Engineering Mathematics, Khanna Publishers, 35th Edition, 2000.

3. Multivariable Calculus

Module 3a**: **Multivariable Calculus (Differentiation)(Prerequisite 2b) (10 hours)
Limit, continuity and partial derivatives, directional derivatives, total derivative; Tangent plane and normal line; Maxima, minima and saddle points; Method of Lagrange multipliers; Gradient, curl and divergence.

Module 3b**: **Multivariable Calculus (Integration)(Prerequisite 3a) (10 hours)
Multiple Integration: Double integrals (Cartesian), change of order of integration in double integrals, Change of variables (Cartesian to polar), Applications: areas and volumes, Center of mass and

Gravity (constant and variable densities); Triple integrals (Cartesian), orthogonal curvilinear coordinates, Simple applications involving cubes, sphere and rectangular parallelepipeds; Scalar line integrals, vector line integrals, scalar surface integrals, vector surface integrals, Theorems of Green, Gauss and Stokes.

Textbooks/References:

1. G.B. Thomas and R.L. Finney, Calculus and Analytic geometry, 9th Edition, Pearson, Reprint, 2002.
2. Veerarajan T., Engineering Mathematics for first year, Tata McGraw-Hill, New Delhi, 2008.
3. Ramana B.V., Higher Engineering Mathematics, Tata McGraw Hill New Delhi, 11th Reprint, 2010.
4. N.P. Bali and Manish Goyal, A text book of Engineering Mathematics, Laxmi Publications, Reprint, 2010.
5. B.S. Grewal, Higher Engineering Mathematics, Khanna Publishers, 35th Edition, 2000.

4. Matrices and Linear Algebra

Module 4a: Matrices (in case vector spaces is not to be taught) (14 hours)
Algebra of matrices, Inverse and rank of a matrix, rank-nullity theorem; System of linear equations; Symmetric, skew-symmetric and orthogonal matrices; Determinants; Eigenvalues and eigenvectors; Diagonalization of matrices; Cayley-Hamilton Theorem, Orthogonal transformation and quadratic to canonical forms.

Module 4b: Matrices (in case vector spaces is to be taught) (8 hours)
Matrices, vectors: addition and scalar multiplication, matrix multiplication; Linear systems of equations, linear Independence, rank of a matrix, determinants, Cramer's Rule, inverse of a matrix,Gauss elimination and Gauss-Jordan elimination.

Module 4c: Vector spaces (Prerequisite 4b) (10 hours)
Vector Space, linear dependence of vectors, basis, dimension; Linear transformations (maps), range and kernel of a linear map, rank and nullity, Inverse of a linear transformation, rank-nullity theorem, composition of linear maps, Matrix associated with a linear map.

Module 4d: Vector spaces (Prerequisite 4b-c) (10 hours)
Eigenvalues, eigenvectors, symmetric, skew-symmetric, and orthogonal Matrices, eigenbases. Diagonalization; Inner product spaces, Gram-Schmidt orthogonalization.

Textbooks/References:

1. D. Poole, Linear Algebra: A Modern Introduction, 2nd Edition, Brooks/Cole, 2005.
2. V. Krishnamurthy, V.P. Mainra and J.L. Arora, An introduction to Linear Algebra, Affiliated East–West press, Reprint 2005.
3. Erwin Kreyszig, Advanced Engineering Mathematics, 9th Edition, John Wiley & Sons, 2006.
4. Veerarajan T., Engineering Mathematics for first year, Tata McGraw-Hill, New Delhi, 2008.
5. N.P. Bali and Manish Goyal, A text book of Engineering Mathematics, Laxmi Publications, Reprint, 2010.
6. B.S. Grewal, Higher Engineering Mathematics, Khanna Publishers, 35th Edition, 2000.

5. ***Differential equations*** (Prerequisite Sections 2 and 3)

Module 5a: First order ordinary differential equations (6 hours)
Exact, linear and Bernoulli's equations, Euler's equations, Equations not of first degree: equations solvable for p, equations solvable for y, equations solvable for x and Clairaut's type.

Module 5b: Ordinary differential equations of higher orders (Prerequisite 2c, 4a) (8 hours)
Second order linear differential equations with variable coefficients, method of variation of parameters, Cauchy-Euler equation; Power series solutions; Legendre polynomials, Bessel functions of the first kind and their properties.

Textbooks/References:

1. Erwin Kreyszig, Advanced Engineering Mathematics, 9th Edition, John Wiley & Sons, 2006.
2. W. E. Boyce and R. C. DiPrima, Elementary Differential Equations and Boundary Value Problems, 9th Edition, Wiley India, 2009.
3. S. L. Ross, Differential Equations, 3rd Ed., Wiley India, 1984.
4. E. A. Coddington, An Introduction to Ordinary Differential Equations, Prentice Hall India, 1995.
5. E. L. Ince, Ordinary Differential Equations, Dover Publications, 1958.
6. G.F. Simmons and S.G. Krantz, Differential Equations, Tata McGraw Hill, 2007.

Module 5c: Partial Differential Equations – First order(Prerequisite 5a-b) (6 hours)
First order partial differential equations, solutions of first order linear and non-linear PDEs.

Module 5d*: Partial Differential Equations*– **Higher order**(Prerequisite 5b-c) (10 hours)
Solution to homogenous and non-homogenous linear partial differential equations second and higher order by complimentary function and particular integral method.
Flows, vibrations and diffusions, second-order linear equations and their classification, Initial and boundary conditions (with an informal description of well-posed problems), D'Alembert's solution of the wave equation; Duhamel's principle for one dimensional wave equation.
Separation of variables method to simple problems in Cartesian coordinates. The Laplacian in plane, cylindrical and spherical polar coordinates, solutions with Bessel functions and Legendre functions. One dimensional diffusion equation and its solution by separation of variables.
Boundary-value problems: Solution of boundary-value problems for various linear PDEs in various geometries.

1. S. J. Farlow, Partial Differential Equations for Scientists and Engineers, Dover Publications, 1993.
2. R. Haberman, Elementary Applied Partial Differential equations with Fourier Series and Boundary Value Problem, 4th Ed., Prentice Hall, 1998.
3. Ian Sneddon, Elements of Partial Differential Equations, McGraw Hill, 1964.
4. Manish Goyal and N.P. Bali, Transforms and Partial Differential Equations, University Science Press, Second Edition, 2010.

6. *Complex Variables*(Prerequisite 2a-c)

Module 6a: Complex Variable – Differentiation(Prerequisite 2a-c) (8 hours):
Differentiation, Cauchy-Riemann equations, analytic functions, harmonic functions, finding harmonic conjugate; elementary analytic functions (exponential, trigonometric, logarithm) and their properties; Conformal mappings, Mobius transformations and their properties.

Module 6b: Complex Variable - Integration(Prerequisite 6a) (8 hours):
Contour integrals, Cauchy-Goursat theorem (without proof), Cauchy Integral formula (without proof), Liouville's theorem and Maximum-Modulus theorem (without proof); Taylor's series, zeros of analytic functions, singularities, Laurent's series; Residues, Cauchy Residue theorem (without proof), Evaluation of definite integral involving sine and cosine, Evaluation of certain improper integrals using the Bromwich contour.

Module 6c: Applications of complex integration by residues: (Prerequisite 2a, 6b) (4 hours)
Evaluation of definite integral involving sine and cosine. Evaluation of certain improper integrals using the Bromwich contour.

Textbooks/References:

1. Erwin kreyszig, Advanced Engineering Mathematics, 9th Edition, John Wiley & Sons, 2006.
2. J. W. Brown and R. V. Churchill, Complex Variables and Applications, 7th Ed., Mc-Graw Hill, 2004.
3. Veerarajan T., Engineering Mathematics for first year, Tata McGraw-Hill, New Delhi, 2008.
4. N.P. Bali and Manish Goyal, A text book of Engineering Mathematics, Laxmi Publications, Reprint, 2010.
5. B.S. Grewal, Higher Engineering Mathematics, Khanna Publishers, 35th Edition, 2000.

7. Numerical Methods

Module 7a: Numerical Methods–1 (Prerequisite 2a)(12 hours)
Solution of polynomial and transcendental equations – Bisection method, Newton-Raphson method and Regula-Falsi method. Finite differences, Relation between operators,Interpolation using Newton's forward and backward difference formulae. Interpolation with unequal intervals: Newton's divided difference and Lagrange'sformulae. Numerical Differentiation, Numerical integration: Trapezoidal rule and Simpson's 1/3rd and 3/8 rules.

Module 7b: Numerical Methods – 2 (Prerequisite 7a, 5a-d)(10 hours)
Ordinary differential equations: Taylor's series, Euler and modified Euler's methods. Runge-Kutta method of fourth order for solving first and second order equations. Milne's and Adam's predicator-corrector methods.
Partial differential equations: Finite difference solution two dimensional Laplace equation and Poission equation, Implicit and explicit methods for one dimensional heat equation (Bender-Schmidt and Crank-Nicholson methods), Finite difference explicit method for wave equation.

Textbooks/References:

1. P. Kandasamy, K. Thilagavathy, K. Gunavathi, Numerical Methods, S. Chand & Company, 2nd Edition, Reprint 2012.

2. S.S. Sastry, Introductory methods of numerical analysis, PHI, 4th Edition, 2005.
3. Erwin kreyszig, Advanced Engineering Mathematics, 9th Edition, John Wiley & Sons, 2006.
4. B.S. Grewal, Higher Engineering Mathematics, Khanna Publishers, 35th Edition, 2010.

8. Transform Calculus

***Module 8a: Transform Calculus*-1** (Prerequisite 2c, 5b-d, 6b) (10 hours)
Polynomials–Orthogonal Polynomials–Lagrange's, Chebysev Polynomials; Trigonometric Polynomials;

Laplace Transform, Properties of Laplace Transform, Laplace transform of periodic functions. Finding inverse Laplace transform by different methods, convolution theorem. Evaluation of integrals by Laplace transform, solving ODEs and PDEs by Laplace Transform method.

***Module 8b:Transform Calculus*-2** (10 hours)
Fourier transforms, Z-transform and Wavelet transforms:properties, methods, inverses and their applications.

Textbooks/References:

1. Erwin kreyszig, Advanced Engineering Mathematics, 9th Edition, John Wiley & Sons, 2006.
2. N.P. Bali and Manish Goyal, A text book of Engineering Mathematics, Laxmi Publications, Reprint, 2010.
3. B.S. Grewal, Higher Engineering Mathematics, Khanna Publishers, 35th Edition, 2000.
4. Veerarajan T., Engineering Mathematics, Tata McGraw-Hill, New Delhi, 2008.

9. Discrete Mathematics

***Module 9a: Sets, relations and functions*:** (8 hours)
Basic operations on sets, Cartesian products, disjoint union (sum), and power sets. Different types of relations, their compositions and inverses. Different types of functions, their compositions and inverses.

Module 9b:Propositional Logic: (6 hours)
Syntax and semantics, proof systems, satisfiability, validity, soundness, completeness, deduction theorem, etc. Decision problems of propositional logic. Introduction to first order logic and first order theory.

Module 9c:Partially ordered sets: (6 hours)
Complete partial ordering, chain, lattice, complete, distributive, modular and complemented lattices. Boolean and pseudo Boolean lattices.

Module 9d:Algebraic Structures: (6 hours)
Algebraic structures with one binary operation – semigroup, monoid and group. Cosets, Lagrange's theorem, normal subgroup, homomorphic subgroup. Congruence relation and quotient structures. Error correcting code. Algebraic structures with two binary operations- ring, integral domain, and field. Boolean algebra and boolean ring (Definitions and simple examples only).

Module 9e:Introduction to Counting:(6 hours)
Basic counting techniques – inclusion and exclusion, pigeon-hole principle, permutation, combination, summations. Introduction to recurrence relation and generating functions.

Module 9f:Introduction to Graphs: (8 hours)
Graphs and their basic properties – degree, path, cycle, subgraph, isomorphism, Eulerian and Hamiltonian walk, trees.

Textbooks/References:

1. C. L. Liu, Elements of Discrete Mathematics, 2nd Ed., Tata McGraw-Hill, 2000.
2. R. C. Penner, Discrete Mathematics: Proof Techniques and Mathematical Structures, World Scientific, 1999.
3. R. L. Graham, D. E. Knuth, and O. Patashnik, Concrete Mathematics, 2nd Ed., Addison-Wesley, 1994.
4. K. H. Rosen, Discrete Mathematics and its Applications, 6th Ed., Tata McGraw-Hill, 2007.
5. J. L. Hein, Discrete Structures, Logic, and Computability, 3rd Ed., Jones and Bartlett, 2010.
6. N. Deo, Graph Theory, Prentice Hall of India, 1974.
7. S. Lipschutz and M. L. Lipson, Schaum's Outline of Theory and Problems of Discrete Mathematics, 2nd Ed., Tata McGraw-Hill, 1999.
8. J. P. Tremblay and R. P. Manohar, Discrete Mathematics with Applications to Computer Science, Tata McGraw-Hill, 1997.

10. *Probability Theory and* Statistics

Probability Theory

Module 10a: **Basic Probability:** (12 hours)
Probability spaces, conditional probability, independence; Discrete random variables, Independent random variables, the multinomial distribution, Poisson approximation to the binomial distribution, infinite sequences of Bernoulli trials, sums of independent random variables; Expectation of Discrete Random Variables, Moments, Variance of a sum, Correlation coefficient, Chebyshev's Inequality.

Module 10b: **Continuous Probability Distributions:** (4 hours)
Continuous random varibales and their properties, distribution functions and densities, normal, exponential and gamma densities.

Module 10c: **Bivariate Distributions:** (4 hours)
Bivariate distributions and their properties, distribution of sums and quotients, conditional densities, Bayes' rule.

Textbooks/References:

1. Erwin Kreyszig, Advanced Engineering Mathematics, 9th Edition, John Wiley & Sons, 2006.
2. P. G. Hoel, S. C. Port and C. J. Stone, Introduction to Probability Theory, Universal Book Stall, 2003 (Reprint).
3. S. Ross, A First Course in Probability, 6th Ed., Pearson Education India, 2002.
4. W. Feller, An Introduction to Probability Theory and its Applications, Vol. 1, 3rd Ed., Wiley, 1968.

Statistics

Module 10d: Basic Statistics: (8 hours)
Measures of Central tendency: Moments, skewness and Kurtosis - Probability distributions: Binomial, Poisson and Normal - evaluation of statistical parameters for these three distributions, Correlation and regression – Rank correlation.

Module 10e: Applied Statistics: (8 hours)
Curve fitting by the method of least squares- fitting of straight lines, second degree parabolas and more general curves. Test of significance: Large sample test for single proportion, difference of proportions, single mean, difference of means, and difference of standard deviations.

Module 10f: Small samples: (4 hours)
Test for single mean, difference of means and correlation coefficients, test for ratio of variances - Chi-square test for goodness of fit and independence of attributes.

Textbooks/References:

1. Erwin Kreyszig, Advanced Engineering Mathematics, 9th Edition, John Wiley & Sons, 2006.
2. N.P. Bali and Manish Goyal, A text book of Engineering Mathematics, Laxmi Publications, Reprint, 2010.
3. B.S. Grewal, Higher Engineering Mathematics, Khanna Publishers, 35th Edition, 2000.
4. Veerarajan T., Engineering Mathematics (for semester III), Tata McGraw-Hill, New Delhi, 2010.

Committee and Area Experts

Experts appointed by AICTE:

- Pankaj Jalote, Distinguished Professor and founding Director, IIIT-Delhi (Chair)
- Manoj Singh Gaur, Director IIT Jammu
- Nutan Limaye, IIT Bombay
- Ramkumar, Pro Vice-Chancellor at Krea University
- Dheeraj Sanghi, Vice-Chancellor, JK Lakshmipat University, Jaipur
- Amit Aggarwal, NASSCOM

Other Experts in the Committee:

- Kishore Kothapalli, Professor, IIIT Hyderabad
- Sudeshna Sarkar, Professor, IIT Karaghpur
- Sukumar Nandi, Professor, IIT Gauhati
- Suchismita Roy, Professsor, NIT Durgapur
- Ashalatha Nayak, Professor, Manipal Institute of Technology
- RBV Subramanyam, Professor, NIT Warangal
- Sanjiva Prasad, Professor, IIT Delhi
- Venkatesh R, TCS Pune
- Viraj Kumar, ACM India Education Committee
- Vishram Thatte, Amazon India
- Vinnie Jauhari, Microsoft India
- R Latha, IBM India
- Gaurav Aggarwal, Google India
- Vinayaka Ram Gururajan, TCS
- Thirumala and Sundar K S, Infosys
- P.B. Kotur, Wipro
- Ishvinder Singh, Cisco Systems, Inc.
- Rahul Suresh Ghali, Accenture

Expert Committees for Different Courses and Micro specializations.

Course	Experts in the Committee
Data Structures	Madhavan Mukund (CMI), Manindra Agrawal (IIT Kanpur), Naveen Garg (IIT Delhi), Amit Kumar (IIT Delhi), Venkatesh Raman (IMSC)
Discrete Mathematics	Nitin Saxena (IIT Kanpur), Somenath Biswas (IIT Goa), Partha Mukhopadhyay (CMI), Bhabani P Sinha (ISI Calcutta)
Algorithm Design and Analysis	Madhavan Mukund (CMI), Manindra Agrawal (IIT Kanpur), Naveen Garg (IIT Delhi), Amit Kumar (IIT Delhi), Venkatesh Raman (IMSC)
Computer Architecture/Organization	M. Balakrishnan (IIT Delhi) , John Jose (IIT Guwahati), Biswabandan Panda (IIT Bombay), Anupam Basu (IIT Kharagpur), Indranil Sengupta (IIT Kharagpur), Anshul Kumar (IIT Delhi)
Advanced Programming	Kishore Kothapalli (IIIT Hyderabad), Vivek Kumar (IIIT Delhi), Swarnendu Biswas (IIT Kanpur)
Operating Systems	Purushottam (Puru) Kulkarni (IIT Bombay), Debadatta Mishra (IIT Kanpur), Chester Rebeiro (IIT Madras)
Databases	Arnab Bhattacharya (IIT Kanpur), Sreenivasa Kumar (IIT Madras), Vikram Goel (IIIT-Delhi), S Sudarshan (IIT Bombay)
Computer Networks	BN Jain (IIIT Delhi), Timothy Gonsalves (IIT Mandi), Vinay Ribero (IITB), Mythili Vutukury (IIT Bombay), PM Sreelakshmi (IIT Mandi)

Machine Learning	CV Jawahar (IIIT Hyderabad), PJ Narayanan (IIIT Hyderabad), Vineeth Balasubramanian (IIT Hyderabad), P. K. Biswas (IIT Kharagpur), Mitesh Khapra (IIT Madras), Piyush Rai (IIT Kanpur), Preethi Jyothi (IIT Bombay), Pabitra Mitra (IIT Kharagpur), Chetan Arora (IIT Delhi)
Security	Sandeep Shukla (IIT Kanpur), Sambuddho Chakravarty (IIIT Delhi), S.Venkatesan (IIIT Allahabad), Vinod Ganapathy (IISc)
Theory of Computing	Raghunath Tewari (IIT Kanpur), Jayalal Sarma (IIT Madras), S Akshay (IIT Bombay)
Compilers	Uday P. Khedker (IIT Bombay), V. Krishna Nandivada (IIT Madras), Dibyapran Sanyal (Nvidia)
Software Engineering Micro specialization	Pankaj Jalote (IIIT-Delhi), Raghu Reddy (IIIT Hyderabad), Vinay Kulkarni (TCS), Meenakshi Dsouza (IIIT Bangalore)
Cyber Security Micro specialization	C. V. Jawahar (IIIT Hyderabad), P. J. Narayanan (IIIT Hyderabad)
Distributed and Cloud Micro specialization	Yogesh Simmhan (IISc Bangalore), Purushottam Kulkarni (IIT Bombay)
HCI micro specialization	Pushpendra Singh (IIIT-Delhi), Anirudha Joshi (IIT Bombay)
Subcommittee for recommendations for online offerings	Dheeraj Sanghi (JKLU), Kishore Kothapalli (IIITH), Ashalata Nayak (MAHE-MIT), Suchismita Roy (NIT Durgapur), RBV Subramanyam (NITW), Divya Bansal (PEC), M. Balakrishnan (IIT Delhi)
Subcommittee for multiple exit options	Prof. M. Balakrishnan (IIT Delhi), Prof. Anshul Kumar (IIT Delhi), Prof. Pankaj Jalote (IIIT-Delhi)

ALL INDIA COUNCIL FOR TECHNICAL EDUCATION

Nelson Mandela Marg, Vasant Kunj, New Delhi 110070

www.aicte-india.org