Curriculum Structure & Syllabi

(As per National Education Policy 2020)

of

B. Tech.

in

Computer Science & Engineering

(w.e.f. 2024-25)

Vision

Mission

Program Educational Objectives

Program Outcomes

Program Specific Outcomes

Overall Credit Structure

Curriculum

Syllabus

Offered By

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
MADAN MOHAN MALAVIYA UNIVERSITY OF TECHNOLOGY
(MMMUT) GORAKHPUR-273 010, UP, INDIA
JUNE 2025

B. Tech. in Computer Science and Engineering

Vision

To become a leader of education, research, and innovation in Computer Science and Engineering and to produce under graduates who are globally recognized as innovative and well-prepared computing professionals.

Mission

1. To create, share and disseminate knowledge through research and education in the theory and application of computing
2. To train the students in different aspects of computing discipline for enhancing, augmenting, and updating their technical skills
3. To inculcate the spirit of analysis, teamwork, innovation, and professionalism among the students

Programme Educational Objectives (PEO)

PEO-1 To inculcate the knowledge of the fundamentals of the mathematics, science & engineering disciplines for developing the ability to formulate, solve and analyze the problems of Computer Science & Engineering field and to provide them the skills for the pursuit of under-graduate studies, research and development and higher education.

PEO-2 To provide an understanding of the prerequisite of the software, technical aspects, and design for coming up with novel engineering solutions and efficient product developments.

PEO-3 To assist the students in the pursuit of a successful career by adopting ethical practices and social responsibility.

PEO-4 To provide students with the technical as well as soft skills required by the national as well as international organizations.

PEO-5 To elevate cognizance in the students toward unending learning and to inculcate ethical and moral ways.

PEO-6 To give students the knowledge of the contemporary technologies, practical experiences, and possibilities in the field of Computer Science & Engineering and to provide the multidisciplinary knowledge to develop the team spirit and leadership qualities by working on multidisciplinary projects.

Programme Outcome (POs)

PO-1 The students will develop the ability towards the application of fundamental knowledge of computing, mathematics, algorithms and computer science & engineering precepts and rationales for developing the solutions of critical engineering problems. (Rudimentary engineering analytical skills).

PO-2 The under graduating students will be able to model and carry out the experiments by using the fundamental knowledge of computer science & engineering discipline and derive the conclusions by analyzing and interpreting the data.

PO-3 The students will be able to analyze, design, implement and assess a computer-based information system, procedure, module, or program to fulfil the requirements along with the consideration of economic, social, privacy and reliability constraints. (Innovative skills)

PO-4 The students will be able to perform efficaciously in multi-disciplinary teams. (Team spirit)

PO-5 The students will develop the analytical skills to critically analyze, recognize, formulate, and devise solutions to the engineering problems by using adequate computing and engineering skills and knowledge. (Engineering problem solving skills)

PO-6 The students will have the awareness towards the professional, ethical practices, legal, security & social consequences, and obligation. (Professional integrity).

PO-7 The students will have efficient speaking skills and written/interpersonal communication skills. (Oral & written communication skill)

PO-8 To impart exhaustive education to the students required to understand and analyze the local and global consequences of computer science & engineering solutions ranging from individuals and organizations to society. (Engineering consequences assessment skills)

PO-9 The students will develop the realization of the requirement of and the ability to indulge in maintaining professional growth and unending learning. (Continuing education cognizance).

PO-10 The students will have the cognition towards the current issues and problems. (Societal awareness)

PO-11 The students will possess the ability to utilize the knowledge of innovative computing equipment's required for engineering tasks. (Pragmatic skills)

PO-12 The students will be able to apply the design and evolution precepts in the development of software and hardware computer systems of variable complications. (Software hardware interface).

Programme Specific Outcome (PSOs)

PSO1. Ability to be lifelong learner to adapt innovation.

PSO2. Ability to learn the best practices regarding ideating, innovating and to be able to attain successful career with globally employable capabilities.

PSO3. Ability to be open to international cultures and demands.

SYLLABUS AND CREDIT STRUCTURE FOR B. TECH. (CSE)
(SESSION 2024-2025 AND ONWARDS)
OVERALL CREDIT STRUCTURE FOR B.TECH. (CSE)

<table>
<tr><th colspan="4">Credit Courses</th></tr>
<tr><th colspan="2">Core Courses (CC)</th><th colspan="2">Electives Courses (EC)</th></tr>
<tr><th>Category</th><th>Min. Credits</th><th>Category</th><th>Min. Credits</th></tr>
<tr><td>Basic Sciences & Maths (BSM)</td><td>20</td><td rowspan="3">Professional Electives (PE)/ Open Electives (OE)</td><td rowspan="3">36</td></tr>
<tr><td>Engineering Fundamentals (EF)</td><td rowspan="2">24</td></tr>
<tr><td>Professional Skill (PS)</td></tr>
<tr><td>Professional Core (PC)</td><td>48</td><td>Humanities & Social Science Elective (HSSE)</td><td>04</td></tr>
<tr><td>Management (M)</td><td>04</td><td></td><td></td></tr>
<tr><td>Humanities & Social Science (HSS)</td><td>08</td><td></td><td></td></tr>
<tr><td>Minor Project (P)</td><td>06</td><td></td><td></td></tr>
<tr><td>Industrial Practice (IP) (In Industry)/ Major Project (MP) (In University)</td><td>10</td><td></td><td></td></tr>
<tr><td>Sub-total</td><td>120</td><td>Sub-total</td><td>40</td></tr>
<tr><td>Grand Total</td><td>160</td><td colspan="2"></td></tr>
<tr><th colspan="4">Non-Credit Courses</th></tr>
<tr><td colspan="3">One Expert Lecture per semester for students (Mandatory).
(BSM-Ist year), (PC-2nd Year), (T&P-3rd Year)</td><td>Non-Credit</td></tr>
<tr><td colspan="3">Social work/Training of at least 60 hours during break after first/ second semester (Mandatory) (Dean of Extension, Field Outreach and Alumni Relations).</td><td>Non-Credit</td></tr>
<tr><td colspan="3">Industrial Training during the summer break after fourth semester (Mandatory).</td><td>Non-Credit</td></tr>
<tr><td colspan="3">One -week workshop during the winter break after fifth semester on professional/ industry/ Social/ entrepreneurial orientation (Mandatory) (Dean of Extension, Field Outreach and Alumni Relations).</td><td>Non-Credit</td></tr>
<tr><td colspan="3">Value Added Courses (VAC) / Audit Courses (AC)
Two of the Value-Added Courses / Audit Courses are compulsory.</td><td>Non-Credit</td></tr>
<tr><td colspan="3">Extracurricular Activities Courses (ECA)
Two compulsory courses from the following S. No (ii) to (v) non-credit courses:
(i) Induction Program (compulsory)
(ii) Skill development
(iii) Unity and Discipline (NCC or NSS)
(iv) Sports, Cultural and Games
(v) Personality Development</td><td>Non-Credit</td></tr>
<tr><td colspan="4">Minor Degree (MD) from any Department and Micro Specializations (MS) within the Department</td></tr>
<tr><td colspan="3">• The total number of credits for graduation will be kept to minimum 160. The additional 18-20 credits required for Minor Degree Courses.</td><td>Offered as a Professional Electives (PE)</td></tr>
</table>

• Micro specializations (MS) will be run by the department in order to aligned to industry careers or higher studies	

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
MADAN MOHAN MALAVIYA UNIVERSITY OF TECHNOLOGY (MMMUT)
GORAKHPUR-273 010, UP, INDIA

SEMESTER WISE CREDIT STRUCTURE FOR B. TECH. (CSE)

Category/Semesters	**I**	**II**	**III**	**IV**	**V**	**VI**	**VII**	**VIII**	**Total**
Basic Sciences & Maths (BSM)	8	8	0/4	4/0					**20***
Humanities & Social Science (HSS)	4	4							**08***
Humanities & Social Science Elective (HSSE)					4				**04***
Management (M)						4			**04***
Engineering Fundamentals (EF)	4	4	8/4	0/4					**16***
Professional Skill (PS)	4	4							**08***
Professional Core (PC)			12	12	12	12			**48***
Professional Electives (PE)/ Open Electives (OE)				4-8	28-32				**36***
Minor Project (P)						0	6		**06***
Industrial Practice (IP) (in Industry)/ Major Project (MP) (In University)								10	**10***
Total Credit	**20***	**20***	**20***	**20-24***	**16*-32***	**16*-32***	**6-30***	**10-30***	**160***
	80-84*				**76-80***				
Total Courses Offered	**05***	**05***	**05***	**05*-06***	**04*-08***	**04*-08***	**00-06***	**00-05***	**36***

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

First Year, Semester I

S. N.	**Category**	**Paper Code**	**Subject**	**L**	**T**	**P**	**Credit**
1.	BSM	BSM-110	Engineering Mathematics-I	3	1	0	4
2.	BSM	BSM-131/181	Engineering Physics	3	0	2	4
3.	EF	BCS-110/160	Introduction to C Programming	3	0	2	4
4.	PS	BCS-111	Web Designing-1	2	0	4	4
5.	HSS	BHS- 101/151	Universal Human Values: Understanding Harmony	3	1	0	4
			Total	**13-14**	**1-3**	**6-10**	**20**
6.	ECA-I		Induction Program	-	-	-	0

 # Group-1: CSE, IT, CH, CE; Group-2: ECE, ECE(IOT), ME, EE.

First Year, Semester II

S. N.	Category	Paper Code	Subject	L	T	P	Credit
1.	BSM	BSM-160	Engineering Mathematics II	3	1	0	4
2.	BSM	BSM-140/BSM-190	Environmental Science and Green Chemistry	3	0	2	4
3.	EF	BEE- 110 / BEE-160	Basic Electrical Engineering	3	0	2	4
4.	PS	BCS-161	Web Designing-2	2	0	4	4
5.	HSS	BHS- 102/152	Technical Writing and Professional Communication	2	1	2	4
			Total	**13-14**	**1-3**	**6-10**	**20**
6.	VAC/AC	BCS-162	Design Thinking in Information & Understanding	0	0	2	0
7.	ECA-II			**-**	**-**	**-**	**0**

Second Year, Semester III

S. N.	Category	Paper Code	Subject	L	T	P	Credit
1.	EF	BCS-210	EF-3 (Group-1) Discrete Structure	3	1	0	4
2.	EF	BCS-211	EF-4- Digital Logic and Design	3	0	2	4
3.	PC	BCS-212	PC-1 Object Oriented Programming through JAVA	3	0	2	4
4.	PC	BCS-213	PC-2 Theory of Computation	3	1	0	4
5.	PC	BCS-214	PC-3 Principles of Data Structures	3	0	2	4
			Total	**15**	**1-5**	**0-8**	**20**
6.	VAC/AC	AUC-101	Constitution of India	2	0	0	0
7.	ECA-III			-	-	-	0

Second Year, Semester IV

S. N.	Category	Paper Code	Subject	L	T	P	Credit
1.	EF	BSM-212/262	MATH-3 Operational Research	3	1/0	0/2	4
2.	PC	BCS-261	PC-4 Design & Analysis of Algorithms	3	1/0	0/2	4
3.	PC	BCS-262	PC-5 Computer Organization and Architecture	3	1/0	0/2	4
4.	PC	BCS-263	PC-6 Database Management Systems	3	1/0	0/2	4

	Student may choose either PE-1 or PE-2 or Both PE-1 and PE-2.						
5.	PE 1	ECS-103	Introduction to Data Science	3	1	0	4
		ECS-104	Introduction to Artificial Intelligence	3	1	0	4
		ECS-105	Cryptography and Network Security	3	1	0	4
		ECS-106	Computer Graphics for Virtual Reality	3	1	0	4
		ECS-107	Fundamental of Blockchain Technology and Applications	3	1	0	4
		ECS-108	Introduction to C Programming	3	0	2	4
		ECS-109	Global Positioning System	3	1	0	4
6.	PE 2	1. ECS-201 2. ECS-202 3. ECS-203	1. Advanced Programming Techniques 2. Fundamental of Blockchain 3. Software Engineering	3	1/0	0/2	4
			Total	**15-18**	**0-6**	**0-12**	**20-24**
7.	VAC/AC	AUC-102 to AUC-115		2	0	**0**	0
8.	ECA-IV			**-**	**-**	**-**	0

List of Extra Curricular Activity (ECA) Courses

ECA-II						
S. No.	Branch	Category	Subject Name	Subject Code	Hours/ Week	Credit
1.	Open to all Branches	ECA	Skill Development-I	ECA-151	2	0
2.	Open to all Branches	ECA	Unity and Discipline (NCC)-I	ECA-171	2	0
3.	Open to all Branches	ECA	Unity and Discipline (NSS)-I	ECA-172	2	0
4.	Open to all Branches	ECA	Games & Sports-I	ECA-181	2	0
5.	Open to all Branches	ECA	Cultural, Art & Literary-I	ECA-182	2	0

List of Value-Added Courses (VAC)/Audit Courses (AC)

S. No.	**Subjects**	**Codes**
1.	Constitution of India	AUC 101
2.	Indian Culture and Heritage	AUC 102
3.	Indian Architecture	AUC 103
4.	Indian Festivals	AUC 104

5.	Vaidic Mathematics	AUC 105
6.	Astronomy	AUC 106
7.	Arts of India	AUC 107
8.	Intellectual Property Right	AUC 108
9.	Human Rights	AUC 109
10.	Logical Research	AUC 110
11.	Professional Ethics	AUC 111
12.	Environmental Law	AUC 112
13.	Health Law	AUC 113
14.	National Cadet Corps	AUC 114
15.	Basics of Human Health and preventive medicines	AUC 115

SKILLS-ENHANCEMENT COURSES FOR EXIT (COMPUTER SCIENCE & ENGINEERING):

2-Months internship for 6-Credits **OR** Two courses mentioned below of 4 to 6 credits.

A. After First Year: UG Certificate (Engg.).

The candidate should pass the following two additional courses (ITI Level) **OR** any two suitable skill-based courses to qualify for **UG Certificate (Engg.).**

S. N.	Category	Paper Code	Subject	L	T	P	Credit
1.	Skill Enhancement	BCS-154	Problem Solving Through Python Programming	2	0	2	3
2.	Skill Enhancement	BCS-155	Computer Troubleshooting & Maintenance	2	0	2	3

OR

Equivalent skills-enhancement courses from MOOC/SWAYAM.

B. After Second Year: UG Diploma (Engg.).

The candidate should pass the following two additional courses **OR** any two suitable skill-based courses to qualify for **UG Diploma (Engg.).**

S. N.	Category	Paper Code	Subject	L	T	P	Credit
1.	Skill Enhancement	BCS-264	Data Analytics using Python	2	0	2	3
2.	Skill Enhancement	BCS-265	Mobile Application Development	2	0	2	3

Syllabus
First year

BSM-110 Engineering Mathematics I

Course category	:	Basic Sciences & Maths (BSM)
Pre-requisite Subject	:	NIL
Contact hours/week	:	Lecture : 3, Tutorial : 1 , Practical: 0
Number of Credits	:	4
Course Assessment methods	:	Continuous assessment through tutorials, attendance, home assignments, quizzes, practical work, record, viva voce, one Minor test and one Major Theory Examination
Course Objectives	:	The course is aimed to develop the basic mathematical skills of engineering students that are imperative for effective understanding of engineering subjects.
Course Outcomes	:	The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course

1. Use of basic differential operators in various engineering problems.
2. Understand the concepts of limit theory and nth order differential equations and their applications to our daily life
3. Solve linear system of equations using matrix algebra.
4. Know about qualitative applications of Gauss, Stoke's and Green's theorem.
5. To know the applications of double and triple integration in finding the area and volume.
6. To inculcate the habit of mathematical thinking and lifelong learning.

Topics Covered

UNIT-I 9

Differential Calculus: Limit, Continuity and Differentiability, Mean value theorems. Leibnitz theorem, Partial derivatives, Euler's theorem for homogenous function, Total derivative, Change of variable. Taylor's and Maclaurin's theorem. Expansion of function of two variables, Jacobian, Extrema of function of several variables.

UNIT-II 9

Linear Algebra: Symmetric, Skew-symmetric matrices, Hermitian, Skew Hermitian Matrices, orthogonal and unitary matrices and basic properties, linear independence and dependence of vectors, Rank of Matrix, Inverse of a Matrix, Elementary transformation, Consistency of linear system of equations and their solution, Characteristic equation, Eigenvalues, Eigen-vectors, Cayley-Hamilton theorem, Diagonalization of matrices.

UNIT-III 9

Multiple Integrals: Double and triple integrals, change of order of integration, change of variables. Application of multiple integrals to surface area and volume. Beta and Gamma functions, Dirichlet integral.

UNIT-IV 9

Vector Calculus: Gradient, Divergence and Curl. Directional derivatives, line, surface and volume integrals. Applications of Green's, Stoke's and Gauss divergence theorems (without Proofs).

Books & References

1. B.S. Grewal: Higher Engineering Mathematics; Khanna Publishers
2. Erwin kreyszig: Advanced Engineering Mathematics, John Wiley & Sons.
3. R. K. Jain and Iyenger: Advanced Engineering Mathematics, Narosa Publications.
4. B.V. Ramana: Higher Engineering Mathematics, Tata Mc. Graw Hill Education Pvt. Ltd.,

BSM-131/181 ENGINEERING PHYSICS

Course category	:	Basic Sciences and Maths (BSM)
Pre-requisite Subject	:	Physics at 12thStandard
Contact hours/week	:	Lecture : 3, Tutorial : 0 , Practical: 2
Number of Credits	:	4
Course Assessment methods	:	Continuous assessment through tutorials, attendance, home assignments, quizzes, practical work, record, viva voce, one Minor test and one Major Theory Examination
Course Objectives	:	Understanding of the principles and concept of Optics, Quantum Mechanics, Fiber Optics, Electrodynamics and Physics of Advanced Materials.
Course Outcomes	:	The students are expected to be able to demonstrate the following knowledge, skills, and attributes after completing this course.

1. Understand the basics principles of Optics and its applications in Engineering and Technology.
2. Compare and understand the uses of various lasers in different fields of Engineering.
3. Know the knowledge of Optical Fibre and their applications in Photonics.
4. Understand the principles of Quantum Mechanics and their applications in Engineering and Technology.
5. Know the principles of Electrodynamics and their applications in Engineering and Technology.
6. Understand the basic properties of advanced materials and their engineering applications.

UNIT-I: **Optics:** **9**

Interference: Interference of light, Interference in thin films, Newton's rings. Refractive index and wavelength determination.

Diffraction: Fresnel and Fraunhofer class of diffraction. Resultant of n-hormonic waves, single, double and N- slit diffraction, Diffraction grating, Grating spectra, Dispersive power.

Polarization: Phenomena of double refraction, Nicol prism, Production and analysis of plane, circular and elliptical polarized light, Retardation Plate, Polarimeter.

Laser: Spontaneous and stimulated emission of radiation, Population inversion, Concept of 3 and 4 level Laser, Construction and working of Ruby, He-Ne lasers, and laser applications.

UNIT-II : Quantum Mechanics and Fiber Optics: **9**

Quantum Mechanics: de Broglie waves, Davisson-Germer experiment, Concept of Phase and Group velocities, Uncertainty principle and its applications, Derivation of time independent and time dependent Schrodinger wave equations. Postulates of quantum mechanics, Significance of wave function, Application of Schrodinger wave equation for a particle in one dimensional infinite potential well.

Fiber Optics: Fundamentals of optical fiber, Acceptance angle and cone, Numerical aperture, Single and Multi-Mode Fibers, Step index and graded index fiber, Propagation Mechanism in optical fibers.

UNIT-III: Electrodynamics: **9**

Scalar and Vector fields, Gradient, Divergence and curl, Concept of displacement current, Maxwell's equation in differential and integral forms, Physical significance of each equation.

Maxwell's equation in free space, Velocity of electromagnetic wave, Transverse nature of the electromagnetic wave, Poynting vector, Maxwell's equations in dielectric and conducting medium, and skin depth.

UNIT-IV: Physics of Advanced Materials: **9**

Concept of energy bands in solids, Semiconducting materials**,** Concept of direct and indirect band gap in semiconductors, Carrier concentration and conductivity in semiconductors, Optoelectronic Materials, Superconducting Materials**,** Temperature dependence of resistivity in superconducting materials, Effect of magnetic field (Meissner effect), Type I and Type II superconductors, London Equations, BCS theory (Qualitative), Introduction of nanoscience, Nanotechnology and its applications.

EXPERIMENTS

1. To determine the specific resistance of a given wire using Carrey Foster's Bridge.
2. To determine the wavelength of sodium light using Newton's Ring experiment.
3. To determine the wavelength of spectral lines of white light using plane diffraction grating.
4. To determine the specific rotation of cane sugar solution using polarimeter.
5. To study the variation of magnetic field along the axis of current carrying circular coil.
6. To study the Hall's effect and to determine Hall coefficient in n type Germanium.
7. To study the energy band gap of Germanium using four probe method.
8. To determine the height of Tower by Sextant.

Books & References

1. Optics- Ajoy Ghatak, Tata McGraw-Hill
2. Optics- N. Subrahmanyam, Brij Lal, M.N. Avadhanulu, S. Chand
3. Quantum Mechanics: Theory and Applications- Ajoy Ghatak, Tata McGraw-Hill
4. Fiber optics and laser Principles and Applications-Anuradha De, New Age International
5. Optical Fibers and its application as sensors by R. K. Shukla, New Age International.
6. Introduction to Electrodynamics by David J. Griffiths, Pearson
7. Physics of Semiconductor Devices, by S. M. Sze, Wiley

8. Concepts of Modern Physics by Arthur Beiser, Tata MCGraw Hill.
9. Introduction to Solid State Physics by C. Kittel, Wiley.
10. Engineering Physics by B. K. Pandey and S. Chaturvedi, 3e Cengage Learning Pvt. Limited, India.
11. Engineering Physics by H. K. Malik and A. Singh Tata MCGraw Hill.
12. Advanced Practical Physics Vol. I and Vol. II by D. K. Dwivedi, Victorius Publishers, New Delhi.

BCS-110/160 Introduction to C Programming

Course category:	Engineering Fundamental (EF)
Pre-requisite Subject:	NIL
Contact hours/week:	Lecture: 3, Tutorial: 0, Practical: 2
Number of Credits:	4
Course Assessment methods:	Continuous assessment through tutorials, attendance, home assignments, quizzes, practical work, record, viva voce, one Minor test and one Major Theory Examination

Course Objective: The course covers the basics of programming and demonstrates fundamental programming techniques, customs and terms including the most common library functions and the usage of the pre-processor. The salient features of course objectives are given below.

1. To develop C Programs using basic programming constructs
2. To develop C programs using arrays and strings
3. To develop applications in C using functions and structures

Course Outcomes: The students are expected to be able to demonstrate the following knowledge, skills, and attitudes after completing this course.

1. Basic terminology used in computer programming.
2. Programs development in C Language by writing, compiling, and debugging.
3. Design of programs involving simple statements, conditional statements, iterative statements, array, strings, functions, recursion, structure, and union.
4. Difference between call by value and call by reference.
5. Dynamic memory allocations and use of pointers.
6. Basic operations on a file.
7. Basics of dynamic memory.

UNIT-I **9**

Basics of programming: Approaches to Problem Solving, Concept of Algorithm and Flow Charts, Types of Computer Languages: Machine Language, Assembly Language and High-Level Language, Concept of Assembler, Compiler, Linker, and Loader. Data types, Storage Classes: Auto, Extern, Register and Static. Operators, Expressions, Operator Precedence and Associativity.

Fundamentals of C Programming: Structure of C Program, Writing and Executing the First C Program, Components of C Language, Standard I/O, Formatted I/O. Conditional Program Execution: Applying if

and switch Statements, Nesting if and else. Program Loops and Iterations: Use of while, do while and for Loops, Multiple Loop Variables, Use of break and continue Statements, goto Statement.

UNIT-II **9**

Arrays: One Dimensional, Multidimensional Array and Their Applications, Declaration and Manipulation of Arrays.

Strings: String Variable, String Handling Functions, Array of Strings.

Functions*:* Designing Structured Programs, Functions in C, User Defined and Standard Functions, Formal vs. Actual Arguments, Function Category, Function Prototype, Parameter Passing, Recursive Functions. Storage Classes revisited.

UNIT-III **9**

Pointers: Pointer Variable and its Importance, Pointer Arithmetic Pointers and Arrays, Pointer and Character Strings, Pointers and Functions, Array of Pointers, Pointers to Pointers.

Structure: Declaration and Initialization of Structures, Structure as Function Parameters, Structure Pointers.

Union: Declaration and Initialization of Unions, Union as Function Parameters, Union Pointers.

UNIT-IV **9**

Dynamic Memory Allocation: malloc, calloc, realloc, free functions.

File Management: Defining and Opening a File, Closing a File, Input/ Output Operations in Files. The Pre-processor Directives, Macros. Command Line Arguments. Introduction to Graphics Programming.

EXPERIMENTS

1. Write programs to print statements in sequential order using simple printf, scanf input/output functions.
2. Write programs to implement if-else condition (simple as well as nested) on suitable problems.
3. Write a program to implement switch-case conditional logic on suitable examples.
4. Write programs to implement for, while and do-while loop control statements on suitable problems.
5. Write programs to implement 1D & 2D array concepts on suitable problems such as sorting of elements, searching of element, matrix addition, subtraction, multiplication etc.
6. Write programs to implement string related concepts such as sorting of a string, finding its length, reversing, concatenation, comparing two strings etc.
7. Write programs to implement concept of user defined functions (call by value, call by reference, recursive calling etc.) on suitable examples.
8. Write programs to implement concepts of pointer.
9. Write programs to implement the concept of structure and union.
10. Write programs to implement dynamic memory allocation functions (calloc, malloc, free, realloc)
11. Write programs to implement file handling concepts such as reading from a file, writing to a file using file related functions (fclose, fopen, sscanf, sprint, fread, fwrite, getc, putc, getw, putw etc.)

Textbooks

1. Jeri R. Hanly and Elliot B. Koffman, Problem Solving and Program Design in C, 7th Edition, Pearson.

2. Schildt, Herbert, Complete Reference with C, Tata McGraw Hill.
3. Kerninghan and Ritchie, The C programming Language, 2nd Edition, Prentice Hall.
4. Richard Bird, Introduction to Functional Programming using Haskell, 2nd Edition, Prentice- Hall International, 1998.

Reference Books

1. Greg Michaelson, An Introduction to Functional Programming Through Lambda Calculus, Dover Edition, Addition Wesley Publication.
2. Samuel P. Harbison, and Guy L. Steele Jr., C-A Reference Manual, Fifth Edition, Prentice Hall, 2002.

BCS-111 Web Designing-1

Course category:	Professional Skill (PF)
Pre-requisite Subject:	NIL
Contact hours/week:	Lecture: 2, Tutorial: 0, Practical: 4
Number of Credits:	4
Course Assessment methods:	Continuous assessment through tutorials, attendance, home assignments, quizzes, practical work, record, viva voce, one Minor test and one Major Theory Examination

Course Objective: Web designing-I syllabus contains a basic introduction to familiarize students with the basics of designing a website to its tools, software applications and themes. Here are the key topics covered under the introduction to web designing.

1. How to design a website
2. Creating different themes for different layouts
3. How to design the look and feel of a website
4. How to create and design banners, advertisements, etc.

Course Outcomes: The students are expected to be able to demonstrate the following knowledge, skills, and attitudes after completing this course

1. Understand principle of Web page design and about types of websites.
2. Visualize and Recognize the basic concept of HTML and application in web designing.
3. Recognize and apply the elements of Creating Style Sheet (CSS).
4. Understanding the basic concept of Java Script and its application.
5. Understanding the basic concept Angular JS.
6. Learning about the tools and techniques of web design covers using software applications

UNIT-I **6**

HTML: Introduction to web site, Domains and Hosting, Responsive Web Designing, Types of Websites: Static and Dynamic, HTML5, Basic structure of an HTML document, HTML Tags: Heading, Paragraphs, Line Breaks, Text, Lists, Tables, Frames, Hyperlinks, Images, Multimedia, Forms, and their controls.

UNIT-II **6**

Creating Style Sheet (CSS): Creating Style Sheet, CSS Properties, CSS Styling, CSS Id and Class, Box Model, CSS Advanced: Grouping, Dimension, Display, Positioning, Floating, Align, Pseudo class, Navigation Bar, Image Sprites, Attribute sector, CSS Colour. Basics of responsive web design using Bootstrap.

UNIT-III **6**

Java Script: Introduction to Java Script, Variables, Operators, Conditions Statements, Loops, Popup Boxes, Events, Arrays, Objects, Functions in JS, Form Validation and regex, JSON.

UNIT-IV **6**

Angular Java Script: Introduction to AngularJS, MVC Architecture, Expressions and Data Biding, Directives, Controllers, Filters, Forms, Modules, Introduction to Single page application.

EXPERIMENTS

1. To create a simple html file to demonstrate the use of different tags.
2. To create an html file to link to different html page which contains images, tables, and also link within a page.
3. To create an html page with different types of frames such as floating frame, navigation frame & mixed frame.
4. To create a registration form as mentioned below.
5. Procedure: Create an html page named as “registration.html”
    i. set background colors
    ii. use table for alignment
    iii. provide font colors& size
6. To create an html file by applying the different styles using inline, external & internal style sheets.
7. Create a sample HTML form using bootstrap.
8. To write a Javascript program to define a user defined function for sorting the values in an array.
9. To create an html page to explain the use of various predefined functions in a string and math object in java script.
10. To create an html page to explain the use of various predefined functions in a array & Date object in Javascript.
11. To create an html page to demonstrate exception handling in javascript
12. To display the calendar using javascript code by getting the year from the user.
13. To create a html registration form and to validate the form using javascript code.
14. To create a html file. To open new window from the current window using javascript.
15. To create an html page to change the background color for every click of a button using javascript.
16. To create an html page with 2 combo boxes populated with month & year, to display the calendar for the selected month & year from combo box using javascript.
17. To create a html page to display a new image & text when the mouse comes over the existing content in the page.
18. Create a single page application using concepts of Angular JS.

Textbooks:

1. Steven M. Schafer, "HTML, XHTML, and CSS Bible, 5ed", Wiley India
2. Ian Pouncey, Richard York, "Beginning CSS: Cascading Style Sheets for Web Design", Wiley India
3. Douglas Crockford, JavaScript: The Good Parts: The Good Parts. O'Reilly Media, Inc.
4. Brad Green, Shyam Seshadri, AngularJS, O'Reilly Media, Inc

BHS- 101/151 Universal Human Values: Understanding Harmony

Course Category	: HSS
Prerequisite subject	: None
Number of Credits	: 4
Contact Hours/Week	: Lectures: 3, Tutorial: 1, Practical: 0
Course Assessment Methods	: Continuous assessment through tutorials, attendance, home assignments, quizzes, practical work, record, viva voce, one Minor test and one Major Theory Examination
Course Objectives	: The objectives of this course are to: -

1. Develop a holistic perspective in students based on self-exploration about themselves (human being), family, society and nature/existence.
2. Develop understanding (or developing clarity) in students about harmony in the human being, family, society and nature/existence.
3. Strengthen self-reflection in students.
4. Develop commitment and courage in students to act.

Course Outcomes : The students will be able to demonstrate the following knowledge, skills, and attitudes upon completion of the course: -

1. Ability to understand the interconnectedness of humanity and nature as well as the importance of values in interpersonal relationships.
2. Ability to recognize their role as global citizens and understand the importance of actively contributing to the betterment of society through responsible actions.
3. Ability to engage in critical reflection on their own values and beliefs, challenging assumptions and biases to foster personal growth and development.
4. Ability to appreciate and respect diversity thereby promoting communication and conflict resolution skills, promoting dialogue and understanding in resolving interpersonal and intergroup conflicts.

Unit 1 **9**

Introduction to Values: origin, definition, meaning, and types of values; Values in Education System; difference between Values, Morals, and Ethics; Self-Exploration–what is it? - Its content and process; 'Natural Acceptance' and 'Experiential Validation' as the process for self-exploration; Continuous Happiness and Prosperity- A look at basic human aspirations; Right understanding, Relationship and Physical Facility- the basic requirements for fulfilment of aspirations of every human being with their correct priority; Understanding Happiness and Prosperity correctly- A critical appraisal of the current

scenario; Method to fulfil the above human aspirations: understanding and living in harmony at various levels.

Unit 2 **9**

Understanding human being as a co-existence of the sentient 'I' and the material 'Body'; Understanding the needs of Self ('I') and 'Body' - happiness and physical facility; Understanding the Body as an instrument of 'I' (I being the doer, seer and enjoyer); Understanding the characteristics and activities of 'I' and harmony in 'I'; Understanding the harmony of I with the Body: Sanyam and Health; correct appraisal of Physical needs, meaning of Prosperity in detail; Programs to ensure Sanyam and Health.

Unit 3 **9**

Understanding values in human-human relationship; meaning of Justice (nine universal values in relationships) and program for its fulfilment to ensure mutual happiness; Trust and Respect as the foundational values of relationship; Understanding the meaning of Trust; Difference between intention and competence; Understanding the meaning of Respect, Difference between respect and differentiation; the other salient values in relationship; Understanding the harmony in the society (society being an extension of family): Resolution, Prosperity, fearlessness (trust) and co-existence as comprehensive Human Goals; Visualizing a universal harmonious order in society- Undivided Society, Universal Order- from family to world family.

Unit 4 **9**

Understanding the harmony in the Nature; Interconnectedness and mutual fulfilment among the four orders of nature- recyclability and self-regulation in nature; Understanding Existence as Co-existence of mutually interacting units in all-pervasive space; Holistic perception of harmony at all levels of existence; Natural acceptance of human values; Definitiveness of Ethical Human Conduct; Basis for Humanistic Education, Humanistic Constitution and Humanistic Universal Order; Competence in professional ethics.

Text & Reference Books:

1. Andrews, C. (2006). *Slow is beautiful*. New Society Publishers.
2. Gandhi, M. K. (1909). *Hind Swaraj or Indian Home Rule*. Navjeevan Trust.
3. Gandhi, M. K. (2009). *An Autobiography or The Story of My Experiments with Truth* (Mahadev Desai, Trans.). Navjeevan Mudranalay. (Original work published 1925).
4. Gaur, R. R., Sangal, R., & Bagaria, G. P. (2010). *A Foundation Course in Human Values and Professional Ethics*. Excel Books.
5. Govindrajan, M., Senthilkumar, S., & Natarajan, M. S. (2013). *Professional Ethics and Human Values*. Prentice Hall India.
6. Kumarappa, J. C. (2017). *Economy of Permanence.* Sarva Seva Sangh Prakashan.
7. Naagarazan, R. S. (2022). *A Textbook on Professional Ethics and Human Values*. New Age International.
8. Rolland, R. (2010). *Life of Vivekanad* (4th Ed.). Advait Ashram.
9. Schumacher, E. F. (1973). *Small is beautiful. A study of Economics as if people mattered*. Blond & Briggs.
10. Suresh, J., & Raghavan, B. S. (2003). *Human Values and Professional Ethics*. S Chand.

BSM-160		**Engineering Mathematics II**
Course category	:	Basic Sciences & Maths (BSM)
Pre-requisite Subject	:	NIL
Contact hours/week	:	Lecture: 3, Tutorial : 1 , Practical: 0
Number of Credits	:	4
Course Assessment methods	:	Continuous assessment through tutorials, attendance, home assignments, quizzes, practical work, record, viva voce, one Minor test and one Major Theory Examination
Course Objectives	:	The course is aimed to develop the basic mathematical skills of engineering students that are imperative for effective understanding of engineering subjects.
Course Outcomes	:	The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course

1. To solve the ordinary differential equations.
2. To solve the partial differential equations using Lagrange and charpit's method.
3. To solve and understand the properties of Bessel`s and Legendre`s differential equation.
4. Application of partial differential equation in real life problems
5. To solve ODE and PDE with the help of Laplace transform
6. To inculcate the habit of mathematical thinking and lifelong learning.

Topics Covered

UNIT-I 9

Ordinary Differential Equations I: Linear differential equations with constant coefficients ($n^{th} order$), complementary function and particular integral. Simultaneous linear differential equations, solution of second order differential equations by changing dependent and independent variables, Method of variation of parameters, Applications of differential equations to engineering problems

UNIT-II 9

Ordinary Differential Equations II: Series solution of second order differential equations with variable coefficient (Frobeneous method). Bessel and Legendre equations and their series solutions, Properties of Bessel function and Legendre polynomials.

UNIT-III 9

Partial Differential equations: Partial differential equations of the first order, Lagrange's solution, Charpit's general method of solution, Partial differential equations of the second order: Constant coefficient and reducible to constant coefficient, Classification of linear partial differential equations of second order.

UNIT-IV 9

Laplace Transform: Laplace Transform, Laplace transform of derivatives and integrals. Unit step function, Laplace transform of Periodic function. Inverse Laplace transform, Convolution theorem, Applications to solve simple linear and simultaneous differential equations and Partial Differential Equations.

Books & References

1. B.S. Grewal: Higher Engineering Mathematics; Khanna Publishers

2. Erwin kreyszig: Advanced Engineering Mathematics, John Wiley & Sons.
3. R. K. Jain and Iyenger: Advanced Engineering Mathematics, Narosa Publications.
4. B.V. Ramana: Higher Engineering Mathematics, Tata Mc. Graw Hill Education Pvt. Ltd.
5. M.D. Raisinghania, Ordinary and Partial Differential Equations. S Chand Publications.

BSM-140/BSM-190	**Environmental Science and Green Chemistry**
Course category:	Basic Sciences & Maths (BSM)
Pre-requisite Subject:	NIL
Contact hours/week	Lecture : 3, Tutorial : 0 , Practical: 2
Number of Credits:	4
Course Assessment methods:	Continuous assessment through tutorials, attendance, home assignments, quizzes, practical work, record, viva voce, one Minor test and one Major Theory Examination
Course Objectives	Understanding the principles and concepts of Chemistry viz. Chemical Bonding, acidity and basicity, Atmospheric Chemistry & Water Chemistry, Spectroscopic analytical methods and Green Chemistry and solving industrial problems using solid foundation in Chemistry.
Course Outcomes:	The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course

1. To develop the concepts of basic chemistry.
2. To make the students aware of global environmental issues e.g. global warming & Green house effect, Ozone depletion, pollution and its prevention and understand various aspects of atmospheric chemistry.
3. To understand the analytical and conceptual skills required for environmental chemistry research.
4. To understand water treatment for all types of uses and need to protect environment.
5. To understand the specifications of pure water and its purification techniques.
6. To develop the knowledge about Green Chemistry and Green Technology.

Unit 1: **9**

Basic Chemical Concepts

Periodic properties of elements, Ionization potential, electron affinity and electronegativity; mole concept, molarity and normality, Chemical Bonding – MO Theory, MO diagram of diatomic molecules, hydrogen bonding, electrophiles, nucleophiles, inductive effect and mesomeric effect. Reaction Mechanism. Acidity and basicity - Concept of pH.

Unit 2: **9**

Atmospheric chemistry & Water Chemistry

The atmosphere of Earth, layers of atmosphere and temperature inversion, Air pollution, Global warming and Greenhouse effect. Acid rain and Ozone layer depletion. Chemical and photochemical Smog.

Sources of water, conservation of water, impurities in water and their effects. WHO guideline and BIS guideline for drinking water. Hardness of water, Softening of water by Zeolite process, Lime Soda process, Ion exchange process and Reverse osmosis.

Unit 3: **9**

Spectroscopic analytical methods

Absorbance, Transmittance and Beer-lamberts Law. Basic principles of UV-Visible spectroscopy, Fluorescence spectroscopy, Infrared spectroscopy, NMR Spectroscopy. Use of these instrumental techniques for monitoring of environmental pollution.

Environmental problems posed by the use of non-biodegradable polymers widely used in day-to-day life. Incineration as the key method for disposal of polymeric waste. Bio-degradable polymers.

Unit 4: **9**

Green Chemistry

Green Chemistry and Green Technology: New trends in Green chemistry; Green Chemistry Methodologies-Microwave heating, ultrasound technique. Green Chemical Synthesis Pathways; Green reagents, Green solvents.

Experiments:

1. Determination of temporary and permanent hardness in water sample using EDTA as standard solution.
2. Determination of alkalinity in the given water sample.
3. Determination of chloride content in the given water sample by Mohr's method.
4. Determination of percentage of available chlorine in bleaching powder sample.
5. Determination of iron content in the given sample using $K_3[Fe(CN)_6]$ as an external indicator.
6. Determination of Electrical conductivity/TDS of a given water sample using conductivity meter.
7. Determination of dissolved Carbon Dioxide of given water sample.
8. Determination of the biochemical oxygen demand of sewage influent.
9. To calculate the lambda max of the given compound by using UV-Visible spectrophotometer.
10. Determination of nickel / cobalt / copper solutions by UV–visible spectrometry.
11. Examples of Green Synthesis /Reactions.
12. Determination of Turbidity of Water
13. Iodoform test
14. Synthesis of a polymer Bakelite or Polyacrylic acid.

Books & References

1. A Text Book of Environment and Ecology, Shashi Chawla, Tata McGraw Hill
2. Environmental Studies, Raj Kumar Singh, Tata McGraw Hill
3. Engineering Chemistry, Wiley India

4. Engineering Chemistry, Tata McGraw Hill
5. Organic Chemistry, Morrison & Boyd, 6th edition, Pearson Education
6. Fundamentals of Environmental Chemistry, Manahan, Stanley E., Boca Raton: CRC Press LLC.
7. Environment and Ecology, R K Khandal, Wiley India
8. An Introductory Text on Green Chemistry: For Undergraduate Students, lndu Tucker Sidhwani, Rakesh K. Sharma, Wiley
9. A text book of Green Chemistry, Shankar Prasad Deo and Nayim Sepay, Techno World Publication.
10. Introduction to Green Chemistry, John Andraos, Albert S. Matlack, CRC Press

BEE-110/160 **Basic Electrical Engineering**

Course category	:	Engineering Fundamentals (EF)
Pre-requisite Subject	:	NIL
Contact hours/week	:	Lecture: 3, Tutorial: 0, Practical: 2
Number of Credits	:	4
Course Assessment methods	:	Continuous assessment through tutorials, attendance, home assignments, quizzes, practical work, record, viva voce, one Minor test and one Major Theory Examination
Course Objectives	:	**1.** To demonstrate and understand the basic knowledge of electrical quantities such as current, voltage, power, energy, and frequency to understand the impact of technology in a global and societal context. **2.** To demonstrate and understand the basic concepts of analysis of simple DC and AC circuits used in electrical engineering and apply the basic concepts in Electrical engineering for multi-disciplinary tasks.

Course Outcomes: The students are expected to be able to demonstrate the following knowledge, skills, and attitudes after completing this course:

1. Understand the basic properties of electrical elements, and solve problem based on basic electrical circuits & DC network theorems.
2. Understand the fundamental behaviour of AC circuits and solve AC circuit problems.
3. Apply the knowledge gained to explain the behaviour of the circuit at series & parallel resonance of circuit & the effect of resonance.
4. Classify different electrical measuring equipment's and understanding their principles.
5. Understand the basic concepts of magnetic circuits.
6. Explain construction and working principle of transformer.

Topic Covered

UNIT I

D C Circuit Analysis and Network Theorems: **9**

Circuit Concepts: Concepts of network, Active and passive elements, Voltage and current sources, Concept of linearity and linear network, Unilateral and bilateral elements, R, L and C as linear

elements, Source transformation, Kirchhoff's laws, Loop and nodal methods of analysis, Star-delta transformation, Network theorems: Superposition theorem, Thevenin's theorem, Norton's theorem, Maximum Power Transfer theorem.

UNIT II **9**

Introduction to AC Circuits:

AC fundamentals, Analysis of single phase series, parallel and series-parallel RLC Circuits, Resonance in series and Parallel circuit.

Three Phase AC Circuits: Three phase system-its necessity and advantages, Star and delta connections, Balanced supply and balanced load, Line and phase voltage/current relations, three-phase power, and its measurement.

UNIT III **9**

Measuring Instruments:

Fundamentals of measurement & instrumentation, Units, Dimensions and Standards. Error Analysis, types of errors & its analysis. Measuring instruments, construction and working principles of PMMC, Moving Iron and Electro-dynamometer type voltmeters & ammeters, Use of shunts and multipliers.

UNIT IV **9**

Magnetic Circuits and Transformers:

Magnetic circuit concepts, analogy between electric & magnetic circuits, B-H curve, Hysteresis, and eddy current losses.

Single Phase Transformer: Principle of operation, Construction, EMF equation, Power losses, Efficiency, O.C & S.C Test and Introduction to auto transformer.

EXPERIMENTS

1. Verification of Kirchhoff's Law.
2. Verification of Norton's Theorem.
3. Verification of Thevenin's Theorem.
4. Verification of Superposition Theorem.
5. Verification of Maximum Power Transfer Theorem.
6. Verification of Series R-L-C circuit.
7. Verification of Parallel R-L-C circuit.
8. Measurement of Power and Power factor of three phase inductive load by two wattmeter method.
9. To perform O.C. and S.C. test of a single-phase transformer.

Textbooks:

1. Fundamentals of Electric Circuits, C.K. Alexander and M.N.O. Sadiku; TATA McGraw-Hill.
2. Principles of Electrical Engineering, V. Del Toro; Prentice Hall International.
3. Electrical and Electronics Technology, Edward Hughes; Pearson.
4. Basic Electrical Engineering, D P Kothari, I.J. Nagarath; Tata McGraw Hill
5. Electrical Technology, B. L. Thareja and A. K. Thareja; S. Chand.

BCS-161	**Web Designing-2**
Course category:	Professional Skill (PF)
Pre-requisite Subject:	NIL
Contact hours/week:	Lecture: 2, Tutorial: 0, Practical: 4
Number of Credits:	4
Course Assessment methods:	Continuous assessment through tutorials, attendance, home assignments, quizzes, practical work, record, viva voce, one Minor test and one Major Theory Examination

Course Objective: The Advanced Web Design course is designed to prepare students for professional web design work. The class will be a mix of theoretical/soft skills and more practical front-end techniques. The objective is listed below.

1. To design web sites which use HTML tables, forms, frames, and Cascading Style Sheets.
2. To learn the advantages of HTML tables, forms, frames Cascading Style Sheets and CSS box model and when they are best utilized.
3. To provide definitions and explanations for a large number of technical terms and acronyms related to web site design.
4. To apply the techniques and features of imagemaps to web site navigation.
5. To understand the issues related to web graphics (size versus resolution) as well as how to create, optimize, and display graphic images.
6. Be able to create and edit simple animated web graphics.

Course Outcomes: The students are expected to be able to demonstrate the following knowledge, skills, and attitudes after completing this course.

1. Understand principle of Web application design.
2. Understand principle of Application Programming Interface (API).
3. Apply the knowledge of Node JS on web application development.
4. Host the web application on web server.
5. learn how to employ meta tags and HTML cookies to improve the experience of web site visitors.

be able to create, validate, transform and display XML files.

UNIT-I **6**

Introduction to Node JS, Setup Dev Environment, Node JS Modules, Node Package Manager.

UNIT-II **6**

Creating Web server, File System, Debugging Node JS Application, Events, Express.JS, Serving Static Resources.

UNIT-III **6**

Basics of MySQL, Query building, Database connectivity using Node JS

UNIT-IV **6**

Introduction to Nginx server, Introduction to AWS EC2, Web application hosting on AWS.

EXPERIMENTS

1. Create API to authentication using Node JS.
2. Create API for session management in Node JS.
3. Create a web application to search an employee from an employee database.
4. Create a web application to generate salary receipt of an employee of an organization.
5. Create a web application for leave management of employees of an organization.
6. Host the employee web application on AWS EC2 server.

Textbooks:

1. Ethan Brown, Web Development with Node and Express: Leveraging the JavaScript Stack, O'Reilly Media, Inc.
2. David Stokes, MySQL and JSON: A Practical Programming Guide, Oracle Press.

BHS- 102/152 TECHNICAL WRITING AND PROFESSIONAL COMMUNICATION (TW&PC)

Course Category	: HSS
Prerequisite subject	: None
Number of Credits	: 4
Contact Hours/Week	: Lectures: 2, Tutorial: 1, Practical: 2
Course Assessment Methods	: Continuous assessment through tutorials, attendance, home assignments, quizzes, practical work, record, viva voce, one Minor test and one Major Theory Examination
Course Objectives	: The objectives of this course are to: - The course aims- 1. To sensitize the students to understand the role and importance of communication for personal and professional success. 2. To enable the learners to enhance their writing skills in techno-cultural and professional echo-system. 3. To equip learners to differentiate technical writing from general writing. 4. To equip them with technical writing skills. 5. To enable learners to exhibit knowledge, skills, attitude and judgment in and around human communication that facilitate their ability to work collaboratively with others in an interpersonal environment.
Course Outcomes	: The students will be able to demonstrate the following knowledge, skills, and attitudes upon completion of the course: - 1. Overcome the problems she/he shall faces in oral and written communication.

2. Acquire knowledge of and methods for using technical communication, such as reports, proposals, technical letters, etc.
3. Use and Practice compositions correctly.
4. Give presentations in different sessions and make self-appraisal.
5. Learn and understand the various facets of Communication Skills, such as (LSRW) Listening, Speaking, Reading, and writing, and identify, formulate, and solve real-life problems with a positive attitude; also inculcate, the habit of learning and developing communication and soft skills.

Unit 1: Language and Communication **6**

Language Vs communication: Communication as coding and decoding – signs, symbols & pictograph – verbal and non–verbal symbols – Language & communication; Types ofCommunication- functional, situational, verbal, and non-verbal, interpersonal, group, interactive, public, Mass Communication. Thinking and Articulation, critical, creative aspects of articulation.

Skills of Language Acquisition: Natural Language Acquisition Skills: Listening, Speaking, Reading& Writing {LSRW}; Language Acquisition Through Training: Listening, Speaking, Reading, Writing, Grammar & Vocabulary {LSRWGV}

Phrase, Clause & Sentence in Professional Drafting-Simplicity, Clarity and Conciseness of a Presentation, Differentiating between Professional and Creative Writing, Blending of Artistic/Professional Writing, Avoiding gender, racial, and other forms of bias in Professional Writing.Pre-writing, Drafting, and Re-writing.

Unit 2: Towards Technical Writing **6**

Technical Paper Writing: Professional Paper Elements-Front Matter of a Paper, Main Text of a Paper, End Matter of a Paper: Organizing References and Bibliography, Order of a thesis and Paper Elements, Concluding Remarks. **Methods of Research Paper Writing:** Identification of Author and His Writing-Author's name and Affiliation, Joint Authorship of a Paper, Identification of Writing- Title, Keywords, Synopsis, Preface and Abstract. Drafting Research Article & Methodology.

Thesis/Dissertation Writing: Thesis Elements-Front Matter of a Thesis, Main Text of a Thesis, End Matter of a Thesis, Specimen—Thesis and Research Paper, Chapters and Sections-Introductory Chapters and Sections, Statement of the Problems, Plan and Scope, Core Chapters and Sections- Theoretical Analysis and Synthesis, Basic Assumption and Hypothesis.

Professional Presentation & Seminar Delivery Tools: Designing the Presentation; Establishing the Objectives. Making Professional PowerPoint Presentations, Signaling Structure of Presentation through Sentences and Crisp Phrases, Preparing Notes for Professional/Technical Presentation, Text Animation, White Board, Flip Charts, Diagrams, Preparing Cards. Seminar Presentations: Purpose modes and methods. Nascent Emerging Platforms for On-line Presentations viz. Zoom, Webex, Team& Meet etc.

Unit 3: Drafting Skills & Career Correspondence **6**

Professional Drafting: Letters vs. e-mails, Formal and Informal emails, Parts of e-mails, Types of e-mails, Managing tone of E-mails and business Letters, Examples of Letters and E-mail, Professional

Correspondence through E-mail, Job Applications and cover Letters. Introduction to DOs (Demi- Official Letters)

Career & Correspondence: Developing a Professional C.V, Bio Data & Resume. Report Writing, Kinds of Reports, Length of Report, Parts of a Report, Terms of Reference, Collection of Facts, Outlines of Report, Examples of Report, Technical Proposal, Elements of Proposal, Examples of Proposal, drafting of proposal.

Unit 4: Professional Practices with ICT Interface **6**

Conducting Professional Meeting: Pre-meeting Preparation, During Meeting: Action Taken Report (ATR) & New Agenda Points, Post Meeting Follow ups. Notice, Circular, Agenda & Meeting Minutes.

Introduction to Generation–Z, Cyber Identity & Professional Netiquettes for Netizens: DraftingE-mails, Blogs on social media, Videoconferencing. Managing Profiles on social media. What to Write and Share on social media. Telephone Etiquettes & Phubbing.

List of Practical:

1. Introduction to Vowel and Consonant Sounds
2. Monophthongs and Diphthongs
3. Syllable, Word Stress & Intonation
4. Harnessing Non-verbal Communication Skills in Cross-Cultural Environment for the establishment of an ideal Ecosystem to ensure Professional Success
5. Developing Speech, and Proofreading the Same
6. Argumentative Skills & Group Dynamics
7. Preparing CV, Biodata & Resume
8. Types of Interview and Interview Skills
9. GD, PI & Telephonic Interview
10. Presentation Skills, Extempore, Debate and Video Conferencing
11. Netiquettes while Writing Blogs on social media.
12. Ethical Usages of Generative AI

Text / Reference Books

1. Acharya Anita. (2012) Interview Skills- Tips & Techniques. Yking Books, Jaipur.
2. Basu, B. N., (2008) Technical Writing. PHI Learning Pvt. Ltd.., New Delhi.
3. Chauhan, N. K & Singh, S. N. (2013) Formal Letters, Pankaj Publication International, New Delhi.
4. Chhabra T.N. (2018) Business Communication. Sun India Publication New Delhi.
5. Dubey Arjun et.al. (2016) Communication for Professionals. Alfa Publications, Delhi.
6. Gibaldi, Joseph (2021). The MLA Handbook for Writers of Research Papers. Ed. IXth, Modern Language Association of America, NY, US.
7. Gurumani, N. (2010) Scientific Thesis Writing and Paper Presentation, MJP Publishers, Chennai.
8. Hamilton Richard. (2009) Managing Writers. Penguin, India.
9. Mc Graw S. J. (2008) Basic Managerial Skills for All. Ed. 08th, Prentice Hall of India, New Delhi.
10. Murphy & Hildebrandt. (2008) Effective Business Communication. Tata McGraw Hill New Delhi.

11. Pandey, S.P., Singh, S. N. & Kumar, Raman, (2023) Exploring Digital Humanities: Challenges & Opportunities, MacBrain Publishing House, New Delhi.

BCS - 162	**Design Thinking in Information & Understanding**
Course category:	Engineering Fundamental (EF)
Pre-requisite Subject:	NIL
Contact hours/week:	Lecture: 0, Tutorial: 0, Practical: 2
Number of Credits:	1
Course Assessment methods:	Continuous assessment through tutorials, attendance, home assignments, quizzes, practical work, record, viva voce, one Minor test and one Major Theory Examination

Course Objective

The course is designed to give an in-depth understanding on various aspects of innovation, creativity, evolving business models, incubation and entrepreneurship and come up with exposure to design thinking for designing innovative products. By learning how to practice and champion design thinking in any role you're in.

1. Continuously produce breakthrough ideas
2. Catalyze design thinking in your daily routine
3. Unlock the innovative capacity of your team
4. Build a lifelong practice of creative problem-solving

Course Outcomes: The students are expected to be able to demonstrate the following knowledge, skills, and attitudes after completing this course.

1. Explain the fundamentals of Design Thinking and innovation.
2. Analyse the model action plan.
3. Describe the principles of innovation and idea generation for product design.
4. Apply design thinking techniques for given tasks and in solving problems in various sectors.

UNIT-I

Introduction to Design Thinking: Introduction to elements and principles of Design, Importance of Design Thinking, why design thinking skills matter, Fundamental design components. Principles of design. Introduction to design thinking, history of Design Thinking, Design Thinking Framework -Design Thinking Methods - Empathise –Define – Ideate – Prototype – Test- Software Development Methodology.

Design thinking: Design thinking process (empathize, analyze, idea & prototype), implementing the process in driving inventions, design thinking in social innovations. Tools of design thinking - person, customer, journey map, brainstorming, product development.

UNIT-II

Innovation: Art of innovation, Difference between innovation and creativity, role of creativity and innovation in organizations. Creativity to Innovation. Teams for innovation, Measuring the impact and

value of creativity, Innovation Management-Changing Management Paradigms-Design Thinking related to Science and art-Design, Thinking in Business-Linking Design Thinking Solution to Business Challenges
Product Design: Problem formation, Introduction to product design, Product strategies, Product value, Product planning, Product specifications.

UNIT-III
Design thinking for strategic Innovation: An exercise in design thinking – implementing design thinking for better process. Implementation of design thinking process in various Industries. How Is Design Thinking Relevant to Software Development? Design thinking for Startups,

UNIT-IV
Design thinking in various sectors: Case studies in Information Technology, Finance, Education, Management and Retail sector. Analyze and Prototyping, Usability testing, Organizing and interpreting results. Design thinking in relation to computational thinking, Design Thinking Careers and Scope

Textbooks
1. Change by Design, Tim Brown, Harper Bollins (2009)
2. Design Thinking in the Classroom by David Lee, Ulysses Press

Reference Books
1. Design the Future, Shrrutin N Shetty, Norton Press
2. Universal principles of design- William lidwell, kritina holden, Jill butter.
3. The era of open innovation – Chesbrough.H
4. Product Design and Manufacturing, A. K. Chitale and R.C. Gupta, Prentice Hall

SKILL-BASED COURSES TO QUALIFY FOR UG CERTIFICATE (ENGG.) IN COMPUTER SCIENCE & ENGINEERING:

BCS-154	**Problem Solving Through Python Programming**
Course category	UG Certificate (Engg.)
Pre-requisite Subject	NIL
Contact hours/week	Lecture: 2, Tutorial : 0 , Practical: 2
Number of Credits	3
Course Assessment methods	Continuous assessment through attendance, home assignments, quizzes, practical work, record, viva voce, one Minor test and one Major Theory Examination
Course Objectives	1. To introduce the fundamentals of programming and develop logical thinking using algorithms and flowcharts. 2. To familiarize students with Python syntax, data types, input/output operations, and the structure of a Python program. 3. To provide a thorough understanding of Python operators, control structures, and loop constructs for decision making and iterative problem-solving.

4. To develop skills in modular programming using functions, recursive functions, and command-line arguments in Python.
5. To enable students to perform array operations using Python's list and NumPy libraries for scientific computing.
6. To introduce file handling and exception handling techniques for building robust and error-resilient Python applications.

Course Outcomes Upon completion of the course, students will be able to

1. Understand basic programming concepts including algorithms, flowcharts, Python syntax, and data types. Understand basic programming concepts including algorithms, flowcharts, Python syntax, and data types.
2. Apply Python operators and control flow constructs such as decision making and loops in program development.
3. Demonstrate the use of arrays and functions, including recursion and command-line arguments, for modular programming.
4. Implement array manipulations and numerical operations using the NumPy library.
5. Perform file operations including reading, writing, and manipulating text and binary files using Python.
6. Apply exception handling techniques to manage errors and ensure robust program execution.

Topics Covered

UNIT-I 6

Introduction to Programming– Computer Systems, Computer Languages, Algorithms and Flowcharts
Introduction to Python Language: Introduction to Python Language, Features of Python, Comments in Python.
Tokens- Keywords, Identifiers, Constants, Variables, Python Input and Output Statements.
Basic Data Types: int, float, boolean, complex and string and its operations.
Collection Data Types: List, Tuples, Sets and Dictionaries. Data Type conversions.

UNIT-II 6

Operators in Python: Arithmetic operators, Assignment operators, Comparison operators, Logical operators, Identity operators, Membership operators, Bitwise operators, Precedence of operators, Expressions.
Control Flow and Loops: Indentation, if statement, if-else statement, nested if else, chained conditional if- elif -else statement, Loops: while loop, for loop using ranges, Loop manipulation using break, continue and pass.

UNIT-III 6

Arrays: Definition**,** Advantages of Arrays, Creating an Array, Operations on Arrays, and List, Importing the Array Module, Indexing and Slicing on Arrays,

Working with arrays using numPy - Creating arrays using numpy,

Functions: Defining a function, Calling a Function, Passing parameters and arguments, Python Function arguments. Recursive functions, command-line arguments.

UNIT-IV 6

File Handling in Python: Introduction to files, Text files and Binary files, Access Modes, Writing Data to a File-write() and writelines(), Reading Data from a File-read(), readline() and readlines().

Error Handling in Python: Introduction to Errors and Exceptions: Compile-Time Errors, Logical Errors, Types of Exceptions, Python Exception Handling Using try, except and finally statements.

List of Practical:

1. Even or Odd Checker
2. Simple Interest Calculator
3. Palindrome String Checker
4. List Operations (Sum, Max, Search)
5. Student Grade Calculator
6. File Handling: Read and Write a File
7. Simple Class: Bank Account Simulation
8. Calculator using Functions
9. Factorial of a number using loop
10. Find Max, Min, Mean, and Std Deviation

Books & References

1. R.NageswaraRao,"Core Python Programming", Dreamtech.
2. Python Programming: A Modern Approach, Vamsi Kuramanchi,Pearson.
3. Core Python Programming,W.Chun,Pearson.
4. Introduction to Python,Kenneth A. Lambert, Cengage.
5. Learning Python, Mark Lutz,Orielly.

BCS-155	Computer Troubleshooting & Maintenance
Course category	UG Certificate (Engg.)
Pre-requisite Subject	NIL
Contact hours/week	Lecture: 2, Tutorial : 0 , Practical: 2
Number of Credits	3
Course Assessment methods	Continuous assessment through tutorials, attendance, home assignments, quizzes, practical work, record, viva voce, one Minor test and one Major Theory Examination
Course Objectives	1. Understand the structure and function of key computer hardware components including CPU, RAM, ROM, motherboard, and peripheral devices. 2. Demonstrate the ability to configure and update BIOS/UEFI settings and install operating systems and device drivers. 3. Perform disk partitioning, formatting, and system configuration tasks for optimal performance and stability.

4. Identify and resolve common hardware and software problems using diagnostic tools and troubleshooting techniques.
5. Apply preventive maintenance practices to ensure long-term computer functionality and performance.
6. Understand fundamental networking concepts and troubleshoot basic connectivity issues in local and internet-based environments.

Course Outcomes The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course

1. Identify and describe the key hardware components of a computer system including processor, memory, motherboard, and I/O devices.
2. Configure BIOS/UEFI settings and install operating systems along with necessary drivers and external devices.
3. Perform disk management operations such as formatting, partitioning, and defragmentation to optimize system performance.
4. Troubleshoot common hardware and software issues including boot errors, driver conflicts, and system crashes using diagnostic tools.
5. Apply regular maintenance practices including antivirus setup, cleaning, and hardware care to enhance system longevity.
6. Demonstrate basic networking skills including identifying components, setting up small networks, and resolving internet connectivity problems.

Topics Covered

UNIT-I 6

Introduction to PC Hardware:

Introduction to computer components: input/output devices, storage, and memory types (RAM, ROM). Overview of CPU: types and generations (Intel Pentium IV, Dual Core, etc.). Motherboard architecture: components, chipsets, connectors, and slots. Understanding buses, ports (USB, HDMI, VGA), and expansion cards.

UNIT-II 6

BIOS Installation and Configuration:

Introduction to BIOS and UEFI. BIOS settings: boot sequence, security settings, hardware configuration. Installing and configuring operating systems (Windows). Installing and updating device drivers. Disk management: formatting, partitioning, defragmentation. Setting up peripherals: printers, scanners, modems.

UNIT-III 6

Troubleshooting:

Common hardware issues and solutions: boot failures, power issues, overheating. Software issues: driver conflicts, blue screen errors, system crashes. POST (Power-On Self-Test) diagnostics and interpretation. System performance enhancement: antivirus, clean up tools, software updates. Identification and replacement of faulty components (RAM, HDD, motherboard). Preventive maintenance: cleaning, environment control, regular updates.

UNIT-IV 6

Networking Fundamentals:

Basics of networking: LAN, WAN, WLAN, and internet. Network components: switches, routers, hubs, modems. Connecting to the internet: broadband, dial-up, ISDN. Setting up small home/office networks. Troubleshooting network issues: IP conflicts, connectivity loss, DNS errors. Applications of the internet: email, file transfer, cloud computing.

List of Practical

1. Disassemble a personal computer and assemble the same system again. Boot the system and observe the procedure of assembling a computer system.
2. Observe various connectors, ports back and front side of the computer. Write their purpose and specifications. (e.g., Power, PS/2 keyboard and mouse, Serial and parallel, USB, VGA, LAN, Audio & microphone, Firewire, HDMI, games, SATA etc.)
3. Identify BIOS settings, demonstrate starting BIOS, identify how to disable unused devices to decrease security risks. Change booting of computer with different secondary storage CD, HDD, USB etc.
4. Perform low level and high-level formatting of Hard Disk. Format the given Hard Disk using any one technique and create three partitions, two for operation systems and one for data. Install OS of different types. Also search for various data recovery software apply on pen drive/HDD.
5. Observe different types of printers (dot matrix, inkjet & laser, multifunction). Install driver and interface the printers with PC/Laptop on any operating system (connect the printer to one PC directly using USB/Serial/Parallel ports as per the availability; test the functioning of the printer.)
6. Learn the interfacing, installation and working of various devices such as scanner, projector, web cam etc. Connect all these devices with the given PC, install & test them.
7. Recognize common symptoms associated with diagnosing and troubleshooting PCs and utilize Windows built-in diagnostic tools.

Books & References

1. PC Hardware: The Complete Reference – Craig Zacker & John Rourke
2. Troubleshooting, Maintaining and Repairing PCs – Stephen J. Bigelow
3. Computer Networking: Principles, Protocols and Practice – Olivier Bonaventure
4. Manufacturer Manuals & Online Documentation (for BIOS, Drivers, etc.

2nd Year

BCS-210 DISCRETE STRUCTURE
Course Category: EF3
Pre-requisite Subject: NIL
Contact Hours/Week: Lecture: 3, Tutorial: 1 & Practical: 0
Number of Credits: 4

Course Assessment Methods: Continuous Assessment through One Minor Test, Teacher Assessment (Quiz, Tutorial, Assignment, Attendance), Viva-voce, and One Major Theory.

Course Objective: It contains a basic introduction to familiarize students with the basics of software Engineering. Here are the key objectives covered in this course

1. Convert logical statements from informal language to propositional (and quantified) logic expressions.
2. Apply formal methods of symbolic propositional logic, such as calculating validity of formulae and computing normal forms.
3. Use the rules of inference to construct proofs in propositional logic.

Course Outcomes: The students are expected to be able to demonstrate the following knowledge, skills, and attitudes after completing this course.

1. Define, as well as explain with examples, the basic terminology of functions, relations, and sets.
2. Perform the operations associated with sets, functions, and relations.
3. Relate practical examples to the appropriate set, function, or relation model, and interpret the associated operations and terminology in context.
4. Ability to distinguish between the notion of discrete and continuous mathematical structures
5. Ability to construct and interpret finite state diagrams and DFSA
6. Application of induction and other proof techniques towards problem solving

TOPIC COVERED

UNIT-I **9**

Set Theory and Logic - Sets, Functions, Relations, Equivalence Relation,Poset. Functions Logic: Propositional Logic, Truth Tables, Tautologies, Resolution Proof System, Predicate Logic.

UNIT-II **9**

Induction And Combinatorics - Peano's Axioms - Mathematical Induction - Pigeon-Hole Principle - Principle Of Inclusion And Exclusion - Review Of Permutations And Combinations - Distribution Problems - Derangements - Bijection Principle.

UNIT-III **9**

Algebraic Structures- Semi-Groups, Monoids, Groups, Subgroups And Their Properties -Cyclic Groups - Cosets - Permutation Groups - Lagrange's Theorem - Cayley's Theorem -Normal Subgroups - Homomorphism Of Groups - Quotient Groups –Introduction To Rings And Fields.

UNIT-IV **9**

Linear Algebra And Recurrence Relations- Linear Algebra: Vector Space, Basis, Dimension, Orthogonally. Recurrence Relations: Homogeneous And Inhomogeneous Recurrences And Their Solutions - Solving Recurrences Using Generating Functions.

Graph Theory- Definitions And Basic Results - Representation Of A Graph By A Matrix And Adjacency List - Trees - Cycles - Properties - Paths And Connectedness - Subgraphs - Graph Isomorphism - Operations On Graphs - Vertex And Edge Cuts - Vertex And Edge Connectivity.

TEXTBOOKS

1. C.L.Liu and D.P.Mohapatra, " Elements of Discrete Mathematics: A Computer Oriented Approach", Mcgraw Hill, Third Edition, 2012.
2. Kenneth H. Rosen, "Discrete Mathematics and Its Applications" Mcgraw Hill, Seventh Edition, 2012 (Indian Adaptation By Kamala Krithivasan, Iit Madras).

REFERENCE BOOK

1. R. Balakrishnan and K. Ranganathan, "A Text Book Of Graph Theory", Springer
2. Thomas Koshy, "Discrete Mathematics with Applications", Elsevier, 2009.
3. Gary Haggard, John Schlipf, and Sue Whitesides, “Discrete Mathematics for Computer Science”, Cengage Learning Publisher, 2005.
4. B. Bollobás, “Modern Graph Theory”, Springer, New York 1998

BCS-211 DIGITAL CIRCUITS AND LOGIC DESIGN

Course Category: EF4
Pre-requisite Subject: NIL
Contact Hours/Week: Lecture: 3, Tutorial: 0 & Practical: 2
Number of Credits: 4

Course Assessment Method: Continuous Assessment through One Minor Test, Teacher Assessment (Quiz, Tutorial, Assignment, Attendance), Practical Work & Viva-voce, and One Major Theory & One Practical Examinations.

Course Objective: This course helps the students in gaining the knowledge of basic principles of digital circuit design and different number systems. This course helps in building a solid foundation to undertake future courses such as computer organization and architecture.

1. Discuss basic building blocks of logic design
2. Learn how circuits are designed in a real computer system

3. Introduce different number systems
4. Learn current trends in circuit design
5. Make student familiar with issues and trade-offs in the design of digital circuits

Course Outcomes: The students are expected to be able to demonstrate the following knowledge, skills, and attitudes after completing this course.

1. Design a finite state machine and sequential logic design.
2. Synthesize a logic design from a natural language description of a problem.
3. Realize a complete arithmetic and logic unit.
4. Generate a realization of combinational logic in a programmable gate array.
5. Simulate a complete design to evaluate functional correctness and timing.
6. Conduct an experiment to learn the logic design and prototyping process in order to acquire requisite hands-on skills

TOPIC COVERED

UNIT-I **9**

Binary Codes - Weighted and Non-Weighted - Binary Arithmetic Conversion Algorithms - Error Detecting and Error Correcting Codes - Canonical and Standard Boolean Expressions - Truth Tables.

UNIT-II **9**

K-Map Reduction - Don't Care Conditions - Adders / Subtractors- Carry Look-Ahead Adder - Code Conversion Algorithms - Design of Code Converters - Equivalence Functions. Binary/Decimal Parallel Adder/ Subtractor for Signed Numbers - Magnitude Comparator - Decoders / Encoders - Multiplexers / Demultiplexers- Boolean Function Implementation using Multiplexers.

UNIT-III **9**

Sequential Logic - Basic Latch - Flip-Flops (SR, D, JK, T and Master-Slave) - Triggering of Flip-Flops - Counters - Design Procedure - Ripple Counters - BCD and Binary - Synchronous Counters.

UNIT-IV **9**

Registers - Shift Registers - Registers with Parallel Load - Memory Unit - Examples of RAM, ROM, PROM, EPROM - Reduction of State and Flow Tables - Race-Free State Assignment - Hazards.

TEXTBOOKS

1. Morris Mano, Digital Design, Prentice Hall of India, 2001
2. Raj Kamal, Digital Systems Principles and Design, Pearson Education, First Edition, 2007
3. Charles H. Roth, Jr. and Larry L. Kinney, Fundamentals of Logic Design, CL Engineering, Seventh Edition, 2013.

REFERENCE BOOKS

1. W. H. Gothmann, Digital Electronics -An Introduction to Theory and Practice, Prentice Hall of India, 2000
2. Donald D. Givone, Digital Principles and Design, Tata McGraw –Hill, Thirteenth Impression, 2003.

BCS-212 OBJECT ORIENTED PROGRAMMING THROUGH JAVA

Course Category: PC
Pre-requisite Subject: NIL
Contact Hours/Week: Lecture: 3, Tutorial: 0 & Practical: 2
Number of Credits: 4

Course Assessment Method: Continuous Assessment through One Minor Test, Teacher Assessment (Quiz, Tutorial, Assignment, Attendance), Practical Work & Viva-voce, and One Major Theory & One Practical Examinations.

Course Objective: This course helps the students in gaining the knowledge of basic principle of design and analysis of algorithms. This course helps to undertake future courses that assume as a background in data structures and algorithm design.

1. To introduce the object-oriented programming concepts.
2. To understand object-oriented programming concepts and apply them in solving problems.
3. To introduce the principles of inheritance and polymorphism; and demonstrate how they relate to the design of abstract classes.
4. To introduce the implementation of packages and interfaces.
5. To introduce the concepts of exception handling and multithreading.
6. To introduce the design of Graphical User Interface using applets and swing controls

Course Outcomes: The students are expected to be able to demonstrate the following knowledge, skills, and attitudes after completing this course.

1. Able to implement, compile, test and run JAVA programs comprising more than one class and to address a particular software problem.
2. Able to solve real world problems using OOP techniques.
3. To identify different components of client server architecture on Internet computing.
4. Knowledge of how to develop and deploy applications and applets in JAVA.
5. Knowledge of how to develop and deploy GUI using JAVA Swing and AWT.
6. Design, develop and implement interactive web applications.

TOPIC COVERED

UNIT-I **9**

Introduction to the principles of object-oriented programming, Core Java: Introduction, Operator, Data types, Variables, Control Statements, Arrays, Methods & Classes, Constructors, String Handling, Inheritance, Package and Interface.

UNIT-II **9**

Exception Handling, Multithread programming, I/O, Java Applet, Networking, Event handling, Introduction to AWT, AWT controls, Layout managers.

UNIT-III **9**

Java Swing: Creating a Swing Applet, Labels, Text fields, Buttons, Tabbed Panes, JDBC: Connectivity Model, JDBC/ODBC Bridge, JAVA SQL package, connectivity to Remote Database, Remote method invocation (RMI).

UNIT-IV **9**

Java Beans: Application Builder tools, The Bean Developer Kit (BDK), JAR files, Introspection, developing a simple bean, using Bound properties, The Java Beans API, Session Beans, Entity Beans, Introduction to Java Servlet: Servlet Basics, Servlet API basic, Life cycle of a Servlet, Running Servlet.

EXPERIMENTS

1. Basic programs of simple statements, conditional statements, iterative statement, and arrays.
2. Programs having object-oriented concepts like Inheritance and Interface.
3. Programs for Exception Handling and Event Handling.
4. Programs of Threads and Multithreading.
5. Programs related to Applets and Swings.
6. Program including JAVA Beans and Servlets.

TEXTBOOKS

1. Herbert Schildt, Java: The Complete Reference, 9th edition, McGraw Hill Education (India) Pvt. Ltd.
2. T. Budd, Understanding Object-Oriented Programming with Java, Pearson Education
3. Balagurusamy E, "Programming in JAVA", TMH

REFERENCE BOOKS

1. Deitel & Deitel, JAVA: How to Program, Pearson education
2. Margaret Levine Young, "The Complete Reference Internet", TMH.
3. Dustin R. Callway, "Inside Servlets", Addison Wesley.
4. Mark Wutica, "Java Enterprise Edition", QUE.
5. Steven Holzner, "Java2 Black book", Dreamtech.

BCS-213 THEORY OF COMPUTATION

Course Category: PC
Pre-requisite Subject: NIL
Contact Hours/Week: Lecture: 3, Tutorial: 1 & Practical: 0
Number of Credits: 4

Course Assessment Methods: Continuous Assessment through One Minor Test, Teacher Assessment (Quiz, Tutorial, Assignment, Attendance), Practical Work & Viva-voce, and One Major Theory & One Practical Examinations.

Course Objective: This course helps the students in gaining the knowledge of basic concept of machines: finite automata, pushdown automata, linear bounded automata, and Turing machines.

1. Discuss basic theory and applications of finite automata
2. Learn how the concepts introduced in this course are applicable in the design of efficient algorithms
3. Introduce Turing machines as a general model of computation
4. Learn the limits of computation

Course Outcomes: The students are expected to be able to demonstrate the following knowledge, skills, and attitudes after completing this course.

1. Analyse and design finite automata, pushdown automata, Turing machines, formal languages, and grammars.
2. Demonstrate the understanding of key notions, such as algorithm, computability, decidability, and complexity through problem solving.
3. Prove the basic results of the Theory of Computation. State and explain the relevance of the Church-Turing thesis.
4. To solve various problems of applying normal form techniques, push down automata and Turing Machines
5. To Construct context free grammar for various languages
6. To Design Finite Automata's for different Regular Expressions and Languages

TOPIC COVERED

UNIT-I **9**

Alphabets, Strings and Languages, Automata and Grammars, Deterministic Finite Automata (DFA)-Formal Definition, Simplified Notation: State Transition Graph, Transition Table, Language of DFA, Nondeterministic Finite Automata (NFA), NFA with Epsilon Transition, Equivalence of NFA and DFA, Minimization of Finite Automata, Myhill-Nerode Theorem

UNIT-II **9**

Regular Expression (RE), Definition, Operators of Regular Expression and their Precedence, Algebraic Laws for Regular Expressions, Kleen‟s Theorem, Regular Expression to FA, DFA to Regular Expression, Arden Theorem, Non Regular Languages, Pumping Lemma for Regular Languages. Application of Pumping Lemma, Closure Properties of Regular Languages, Decision Properties of Regular Languages, FA with Output: Moore and Mealy Machine, Equivalence of Moore and Mealy Machine, Applications and Limitation of FA.

UNIT-III **9**

Context Free Grammar (CFG) and Context Free Languages (CFL): Definition, Examples, Derivation, Derivation Trees, Ambiguity in Grammer, Inherent Ambiguity, Ambiguous to Unambiguous CFG, Useless Symbols, Simplification of CFGs, Normal Forms for CFGs: CNF and GNF, Closure Proper ties of CFLs, Decision Properties of CFLs: Emptiness, Finiteness and Membership, Pumping Lemma for CFLs. Push Down Automata (PDA): Description and Definition, Instantaneous Description, Language of PDA, Acceptance by Final State, Acceptance by Empty Stack, Deterministic PDA, Equivalence of PDA and CFG, CFG to PDA and PDA to CFG, Two Stack PDA

UNIT-IV **9**

Turing Machines (TM): Basic Model, Definition and Representation, Instantaneous Description, Language Acceptance by TM, Variants of Turing Machine, TM as Computer of Integer Functions, Universal TM, Church‟s Thesis, Recursive and Recursively Enumerable Languages, Halting Problem, Introduction to Undecidability, Undecidable Problems about TMs. Post Correspondence Problem (PCP), Modified PCP, Introduction to Recursive Function Theory.

TEXTBOOKS

1. Micheal Sipser, “Introduction to the Theory of Computation”, Thomson Learning

REFERENCE BOOKS

1. Hopcroft, Ullman, “Introduction to Automata Theory, Languages and Computation”, Pearson Education
2. Peter Linz, "An Introduction to Formal Language and Automata", Narosa Publishing house
3. H R. Lewis and Christos H. Papadimitriou, “Elements of the theory of Computation”, PHI Ltd

BCS-214 PRINCIPLES OF DATA STRUCTURES

Course Category: PC
Pre-requisite Subject: NIL
Contact Hours/Week: Lecture: 3, Tutorial: 0 & Practical: 2
Number of Credits: 4

Course Assessment Methods: Continuous Assessment through One Minor Test, Teacher Assessment (Quiz, Tutorial, Assignment, Attendance), Practical Work & Viva-voce, and One Major Theory & One Practical Examinations.

Course Objective: This course helps the students in gaining the knowledge of basic principles of Data Structures. The principal objectives of this course are:

1. To provide the knowledge of basic data structures and their implementations.
2. To understand importance of data structures in context of writing efficient programs.
3. To develop skills to apply appropriate data structures in problem solving.

Course Outcomes: The students are expected to be able to demonstrate the following knowledge, skills, and attitudes after completing this course.

1. Learn the basic types for data structure, implementation, and application.
2. Know the strength and weakness of different data structures.
3. Use the appropriate data structure in context of solution of given problem.
4. Develop programming skills which require to solve given problem.
5. Apply Algorithm for solving problems like sorting, searching, insertion and deletion of data
6. Solve problem involving graphs, trees and heaps.

TOPICS COVERED

UNIT-I **9**

Introduction: Basic Terminology, Elementary Data Organization, Structure Operations, Complexity and Time-Space Trade-off.

Arrays: Definition, Representation and Analysis, Single and Multi-Dimension Array, Address Calculation, Application of Arrays, Character, String in C, Character String Operation, Arrays Parameters, Ordered List, Sparse Matrices and Vectors

Stacks: Array Representation and Implementation of Stack, Operations on Stacks: Push &Pop, Array Representation of Stack, Linked Representation of Stack, Operations Associated with Stacks, Application of Stack, Conversion of Infix to Prefix and Postfix Expressions, Evaluation of Postfix Expressions using Stack, Application of Recursion in Problem like Tower of Hanoi.

UNIT-II **9**

Queues: Array and Linked Representation and Implementation of Queues, Operations on Queue: Create, Add, Delete, Full and Empty, Circular Queues, D-Queues and Priority Queues.

Linked List: Representation and Implementation of Singly Linked Lists, Two-Way Header List, Traversing and Searching of Linked List, Overflow and Underflow, Insertion and Deletion to / from Linked Lists, Insertion and Deletion Algorithms, Doubly Linked List, Linked List in Array, Polynomial Representation and Addition, Generalized Linked List, Garbage Collection and Compaction.

UNIT-III **9**

Trees: Basic Terminology, Binary Trees, Binary Tree Representation, Algebraic Expressions, Complete Binary Tree, Extended Binary Trees, Array and Linked Representation of Binary Trees, Traversing Binary Trees, Threaded Binary Trees, Traversing Threaded Binary Trees, Huffman Algorithm.

Binary Search Trees: Binary Search Tree (BST), Insertion and Deletion in BST, Complexity of Search Algorithm, Path Length, AVL Trees, B-Trees.

UNIT-IV **9**

Searching and Hashing: Sequential Search, Binary Search, Comparison and Analysis, Hash Table, Hash Functions, Collision Resolution Strategies, Hash Table Implementation.

Sorting: Insertion Sort, Bubble Sorting, Quick Sort, Two Way Merge Sort, Heap Sort, Sorting on Different Keys, Practical Consideration for Internal Sorting.

Graphs: Terminology & Representations, Graphs &Multi-Graphs, Directed Graphs, Sequential Representations of Graphs, Adjacency Matrices, Traversal, Connected Component and Spanning Trees, Minimum Cost Spanning Trees.

EXPERIMENTS

Write C/C++ Programs to illustrate the concept of the following:

1. Implementation of searching and sorting techniques.
2. Implementation of list using array and linked list.
3. Implementation of push and pop operation on stack
4. Implementation of polish notation and its conversion
5. Write a program to solve the problems using iteration/recursion
6. Program for recursion removal using stack
7. Program for insertion /deletion operation on various queue & Implementation of priority queue for process scheduling

8. Program for storing data as tree structure and implementation of various traversal techniques
9. Program for storing data as graph structure and implementation of various traversal techniques
10. Program for finding shortest path in graph

TEXT BOOKS

1. Horowitz and Sahani, Fundamentals of Data Structures, Galgotia Publication, New Delhi.
2. R. Kruse et al, Data Structure and Pragram Design in C, Pearson Education Asia Delhi
3. A. M. Tenenbaum, Data Structures using C & C++, PHI, India
4. K Loudon, Mastering Algorithms with C, Shroff Publication and Distributor Pvt. Ltd.
5. Bruno R Preiss, Data Structure and Algorithms with Object Oriented Design Pattern in C++, John Wiley & Sons
6. Adam Drozdek, "Data Structures and Algorithms in C++", Thomson Asia Pvt. Ltd. Singapore

REFERENCE BOOKS

1. Lewis, H.R., Denenberg, L., Data Structures and their Algorithms. Published by AddisonWesley, UK, 1991
2. Oluwadare, S.A., Agbonifo, O.C., Fundamentals of Data structures and Algorithms. Lecture Notes, 2013

BSM-212/262 OPERATIONAL RESEARCH

Course Category: Basic Sciences & Maths
Pre-requisite Subject: NIL
Contact Hours/Week: Lecture: 3, Tutorial: 1 & Practical: 0
Number of Credits: 4

Course Assessment Methods: Continuous assessment through tutorials, attendance, home assignments, quizzes and One Minor tests and One Major Theory Examination

Course Objective: The course is aimed to develop the mathematical skills and analyzing different situations in the industrial scenario having limited resources and obtain the optimal solution with and without constraints.

Course Outcomes: The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course

1. Identify and develop operational research models from the verbal description of the real system.
2. Able to build and solve Transportation Models and Assignment Models.
3. Knowledge of formulating mathematical models for quantitative analysis of managerial problems in industry
4. Aware with the basic concepts and tools of game theory and can apply these tools to real-life situations.
5. Understand different queuing situations and find the optimal solutions using models for different situations.
6. Able to design new simple models, like: CPM, PERT to improve decision making.

TOPIC COVERED

UNIT-I **9**

Linear Programming (LP): Formulation of mathematical models for various types of L.P. problems, graphical methods of solving L.P. problems, Limitations of L.P. methods. Simplex method, artificial variable technique-the big-M method, two phase Simplex method, revised simplex method, degeneracy, duality in L.P.

UNIT-II **9**

Transportation and Assignment Problems: Transportation model formulation, and solution of transportation problems (Optimal), Assignment model, formulation and solution of assignment problems, sequencing problems.

UNIT-III **9**

Game Theory and Network Techniques: Game theory, solution of games with and without saddle point, rules of dominance, arithmetic, and algebraic methods for 2x2 games, solution of 2xn or m x 2 games. PERT & CPM Models: Characteristics & uses, drawing of network, removal of redundancy in network, computing EST, LFT, critical path, project completion time, Free Slack, Total slack, and independent slack, Project crashing.

UNIT-IV **9**

Queuing Theory: Elements of Queuing model, Pure-birth and Pure-death models, Empirical queuing models – M/M/1: ∞/FCFS, M/M/1: N/FCFS and M/M/C: ∞/FCFS models and their steady state performance analysis.

TEXT BOOKS

1. Hillier, F. S., & Lieberman, G. J. Introduction to operations research- concepts and cases. New Delhi: Tata McGraw Hill (Indian print).
2. Taha, H. A. Operations research-an introduction. New Delhi: Pearson Prentice Hall.
3. Ravindran, A., Phillips, D. T., and Solberg, J. J. Operations research- principles and practice. New Delhi: Wiley India (P.) Ltd. (Indian print).
4. Kanti Swaroop, P K Gupta and Manmohan, Operations Research, Sultan Chand & Sons
5. Gross, D., Shortle, J. F., Thompson, J. M., & Harris, C. M. Fundamentals of queueing theory. Wiley India (P.) Ltd. (Indian print).

BCS-261 **DESIGN & ANALYSIS OF ALGORITHMS**

Course Category: PC
Pre-requisite Subject: NIL
Contact Hours/Week: Lecture: 3, Tutorial: 1/0 & Practical: 0/2
Number of Credits: 4

Course Assessment Methods: Continuous Assessment through One Minor Test, Teacher Assessment (Quiz, Tutorial, Assignment, Attendance), Practical Work & Viva-voce, and One Major Theory & One Practical Examinations.

Course Objective: This course helps the students in gaining the knowledge of basic principle of design and analysis of algorithms. This course helps to undertake future courses that assume as a background in data structures and algorithm design.

1. Discuss basic theory and practice of algorithms
2. Design and implement new algorithms
3. Become familiar with fundamental data structures and with the way these data structures can best be implemented
4. Become accustomed to the description of algorithms in both functional and procedural styles

Course Outcomes: The students are expected to be able to demonstrate the following knowledge, skills, and attitudes after completing this course.

1. Define the basic concepts of algorithms and analyze the performance of algorithms.
2. Discuss various algorithm design techniques for developing algorithms.
3. Discuss various searching, sorting and graph traversal algorithms.
4. Understand NP completeness and identify different NP complete problems.
5. Discuss various advanced topics on algorithm
6. Able to Compare between different data structures and pick an appropriate data structure for a design situation.

TOPIC COVERED

UNIT-I **9**

Introduction: Algorithms, Analyzing Algorithms, Complexity of Algorithms, Growth of Functions, Performance Measurements, Sorting and Order Statistics - Shell Sort, Quick Sort, Merge Sort, Heap Sort, Comparison of Sorting Algorithms, Sorting in Linear Time. Divide And Conquer strategy with Examples such as Sorting, Matrix Multiplication, Convex Hull and Searching.

.

UNIT-II **9**

Greedy Methods with Examples such as Optimal Reliability Allocation, Knapsack, Minimum Spanning Trees – Prim's and Kruskal's Algorithms, Single Source Shortest Paths – Dijkstra's and Bellman Ford Algorithms.

Dynamic Programming with Examples such as Multistage Graphs, Knapsack, All Pair Shortest Paths - Warshal's and Floyd's Algorithms, Resource Allocation problem

UNIT-III **9**

Backtracking, Branch and Bound with Examples such as Travelling Salesman Problem, Graph Coloring, N-Queen Problem, Hamiltonian Cycles and Sum of Subsets

Advanced Data Structures*:* Red-Black Trees, B – Trees, Binomial Heaps, Fibonacci Heaps

UNIT-IV **9**

Selected Topics: String Matching, Text Processing- Justification of Text, Theory of NP-Completeness, Approximation Algorithms and Randomized Algorithms, Algebraic Computation, Fast Fourier Transform

EXPERIMENTS

1. To analyze time complexity of Insertion sort.
2. To analyze time complexity of Quick sort.
3. To analyze time complexity of Merge sort.
4. To Implement Largest Common Subsequence.
5. To Implement Matrix Chain Multiplication.
6. To Implement Strassen's matrix multiplication Algorithm, Merge sort and Quick sort.
7. To implement Knapsack Problem.
8. To implement Activity Selection Problem.
9. To implement Dijkstra's Algorithm.
10. To implement Warshall's Algorithm.
11. To implement Bellman Ford's Algorithm.
12. To implement Naïve String Matching Algorithm.
13. To implement Rabin Karp String Matching Algorithm
14. To implement Prim's Algorithm.
15. To implement Kruskal's Algorithm.

TEXT BOOKS

1. Thomas H. Coreman, Charles E. Leiserson and Ronald L. Rivest, Introduction to Algorithms, PHI
2. RCT Lee, SS Tseng, RC Chang, and YT Tsai, "Introduction to the Design and Analysis of Algorithms", McGraw Hill 2005
3. Ellis Horowitz and Sartaj Sahni, *Fundamentals of Computer Algorithms*, Computer Science Press, Maryland 1978
4. Berman, Paul," Algorithms", Cengage Learning, PHI
5. Aho, Hopcraft, Ullman, "The Design and Analysis of Computer Algorithms" Pearson Education, 2008

REFERENCE BOOKS

1. Berlion, P. Izard, P., Algorithms-The Construction, Proof and Analysis of Programs, 1986. Johan Wiley & Sons
2. Ellis Horowitz, Sartaj Sahni, and Sanguthevar Rajasekaran, *Computer Algorithms*, W. H. Freeman, NY, 1999
3. Goodman, S.E. & Hedetnien, introduction to Design and Analysis of Algorithms, 1997 MGH
4. Knuth, D. E, Fundamentals of Algorithms: The Art of Computer Programming Vol, 1985

BCS-262 COMPUTURE ORGANIZATION & ARCHITECTURE

Course Category: PC
Pre-requisite Subject: NIL
Contact Hours/Week: Lecture: 3, Tutorial: 1/0 & Practical: 0/2
Number of Credits: 4

Course Assessment Methods: Continuous Assessment through One Minor Test, Teacher Assessment (Quiz, Tutorial, Assignment, Attendance), Practical Work & Viva-voce, and One Major Theory & One Practical Examinations.

Course Objectives: This course helps the students in gaining the knowledge of Computer Organization and Architecture. The principal objectives of this course are:

1. To understand the structure, function and characteristics of computer systems.
2. To understand the design of the various functional units and components of computers.
3. To explain the function of each element of a memory hierarchy,
4. To identify and compare different methods for computer I/O.

Course Outcomes: The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course.

1. To study the design of arithmetic and logic unit and implementation of fixed point and floating-point arithmetic operations
2. To study the two types of control unit techniques and the concept of Pipelining
3. To study the hierarchical memory system including cache memories and virtual memory
4. To study the different ways of communicating with I/O devices and standard I/O interfaces
5. Elaborate advanced concepts of computer architecture, Parallel Processing, interprocessor communication and synchronization
6. Describe the operations and language of the register transfer, micro operations and input- output organization.

TOPICS COVERED

UNIT-I **9**

Basics: Functional Blocks in a Computer System, Floating point representation - Precision and Accuracy, Computer Arithmetic, Addressing Modes, Instruction Set and Instruction Execution Flow, Instruction Format, Instruction Set Design, Data Transfer & Manipulations operations, Arithmetic and Logic operations, Shift Micro-Operation.

UNIT-II **9**

Design of Arithmetic Circuits: Addition & Subtraction of Signed Numbers, Multiplication, Integer Division, Fast Adder and Fast multipliers, Floating Point Operations, Decimal Arithmetic Unit, Decimal Arithmetic Operations, Arithmetic and Instruction Pipeline, its Performance.
Control Design: Hardwired control design, Micro Programmed Control Unit, Micro-instruction design

UNIT-III **9**

Characteristics of memory system, Memory hierarchy, Cache Memory- Cache memory principles, Elements of cache design- cache address, size, mapping functions, replacement algorithms, write policy, Internal Memory- Main Memory (RAM and ROM Chips), semiconductor memory, External Memory- Hard Disk organization, RAID, Virtual Memory concept.

UNIT-IV **9**

I/O modules- Module function and I/O module structure, Programmed I/O, Polling I/O, Interrupt driven I/O, DMA function, Synchronous and Asynchronous serial data communication, Computer peripherals like keyboard, mouse, printer, scanner, and display devices.

TEXT BOOKS

1. Computer System Architecture - M. Mano
2. Carl Hamacher, Zvonko Vranesic, Safwat Zaky Computer Organization, McGraw-Hill, Fifth Edition, Reprint 2012
3. John P. Hayes, Computer Architecture and Organization, Tata McGraw Hill, Third Edition, 1998.

REFERENCE BOOKS

1. William Stallings, Computer Organization and Architecture-Designing for Performance, Pearson Education, Seventh edition, 2006.
2. Behrooz Parahami, "Computer Architecture", Oxford University Press, Eighth Impression, 2011.
3. David A. Patterson and John L. Hennessy, "Computer Architecture-A Quantitative Approach", Elsevier, a division of reed India Private Limited, Fifth edition, 2012.

BCS-263 **DATABASE MANAGEMENT SYSTEMS**

Course Category: PC
Pre-requisite Subject: NIL
Contact Hours/Week: Lecture: 3, Tutorial: 1/0 & Practical: 0/2
Number of Credits: 4

Course Assessment Methods: Continuous Assessment through One Minor Test, Teacher Assessment (Quiz, Tutorial, Assignment, Attendance), Practical Work & Viva-voce, and One Major Theory & One Practical Examinations.

Course Objective: This course helps the students in gaining the knowledge of basic principle of Database Management Systems and to undertake future courses that assume as a background in Database Management Systems.

1. Discuss basic theory and practice of database management systems
2. Design and implement new databases
3. Become familiar with fundamentals of database management systems and with the way these can best be implemented
4. Become accustomed to the description of database management systems in both functional and procedural styles

Course Outcomes: The students are expected to be able to demonstrate the following knowledge, skills, and attitudes after completing this course.

1. Define the basic concepts of database management systems.
2. Discuss various database design techniques for developing database systems.
3. Discuss various techniques for having control over concurrent database transactions.

4. Discuss various advanced topics on database management systems
5. Understand various transaction processing, concurrency control mechanisms and database protection mechanisms.
6. Demonstrate an understanding of normalization theory and apply such knowledge to the normalization of a database.

TOPIC COVERED

UNIT-I **9**

Introduction: An Overview of Database Management System, Database System vs File System, Database System Concept and Architecture, Data Model Schema and Instances, Data Independence and Database Language and Interfaces, Data Definitions Language, DML, Overall Database Structure.

Data Modeling using Entity Relationship Model: ER Model Concepts, Notation for ER Diagram, Mapping Constraints, Keys, Concepts of Super Key, Candidate Key, Primary Key, Generalization, Aggregation, Reduction of An ER Diagrams to Tables, Extended ER Model, Relationship of Higher Degree.

UNIT-II **9**

Relational Data Model and Language: Relational Data Model Concepts, Integrity Constraints, Entity Integrity, Referential Integrity, Keys Constraints, Domain Constraints, Relational Algebra, Relational Calculus, Tuple and Domain Calculus.

Introduction on SQL: Characteristics of SQL, Advantage of SQL. SQL Data Type and Literals. Types of SQL Commands. SQL Operators and their Procedure. Tables, Views and Indexes. Queries and Sub Queries. Aggregate Functions. Insert, Update and Delete Operations, Joins, Unions, Intersection, Minus, Cursors, Triggers, Procedures in SQL/PL SQL

UNIT-III **9**

Database Design &Normalization: Functional Dependencies, Normal Forms, First, Second, Third Normal Forms, BCNF, Inclusion Dependence, Loss Less Join Decompositions, Normalization using FD, MVD, and JDS, Alternative Approaches to Database Design.

UNIT-IV **9**

Transaction Processing Concept: Transaction System, Testing of Serializability, Serializability of Schedules, Conflict & View Serializable Schedule, Recoverability, Recovery from Transaction Failures, Log Based Recovery, Checkpoints, Deadlock Handling.

Distributed Database: Distributed Data Storage, Concurrency Control, Directory System.

Concurrency Control Techniques: Concurrency Control, Locking Techniques for Concurrency Control, Time Stamping Protocols for Concurrency Control, Validation Based Protocol, Multiple Granularity, Multi Version Schemes, Recovery with Concurrent Transaction, Case Study of Oracle.

EXPERIMENTS

1. Write SQL queries using SQL operators
2. Write SQL query using character, number, date and group functions

3. Write SQL queries for relational algebra
4. Write SQL queries for extracting data from more than one table
5. Write SQL queries for sub queries, nested queries
6. Write program using PL/SQL
7. Concepts for ROLL BACK, COMMIT & CHECK POINTS
8. Create VIEWS, CURSORS and TRGGERS & write ASSERTIONS.
9. Create FORMS and REPORTS
10. Design of tables by normalization and dependency analysis
11. Writing application software with host language interface

TEXT BOOKS

1. Date C J, An Introduction to Database Systems, Addison Wesley
2. Korth, Silbertz, Sudarshan, Database Concepts, McGraw Hill
3. Elmasri, Navathe, Fundamentals of Database Systems, Addison Wesley
4. O"Neil, Databases, Elsevier Pub.
5. Leon& Leon, Database Management Systems, Vikas Publishing House
6. Bipin C. Desai, An Introduction to Database Systems, Galgotia Publications
7. Majumdar & Bhattacharya, Database Management System, TMH
8. Ramkrishnan, Gehrke, Database Management System, McGraw Hill
9. Kroenke, Database Processing Fundamentals, Design and Implementation, Pearson Education.
10. J. D. Ulman, Principles of Database and Knowledge base System, Computer Science Press.
11. Maheshwari Jain. DBMS: Complete Practical Approach, Firewall Media, New Delhi

REFERENCE BOOKS

1. Ramon a. Mato-Toledo, Pauline K. Cushman, Database Management Systems, Schaums Outline series, TMH, New Delhi Special Indian Edition 2007
2. Ivan Bayross, Mastering Database Technologies, BPB Publications, New Delhi - First Indian Edition 2006, Reprinted 2011

ECS-103 INTRODUCTION TO DATA SCIENCE

Course Category: PE1
Pre-requisite Subject: NIL
Contact Hours/Week: Lecture: 3, Tutorial: 1 & Practical: 0
Number of Credits: 4

Course Assessment methods: Continuous assessment through assignments, quizzes, tutorials and one minor test and one major examination.

Course Objective:

1. To provide the knowledge and expertise to become a proficient data scientist.

2. Demonstrate an understanding of statistics and machine learning concepts that are vital for data science.
3. Produce Python code to statistically analyse a dataset.

Course Outcomes: The students are expected to be able to demonstrate the following knowledge, skills, and attitudes after completing this course.

1. To explain how data is collected, managed, and stored for data science.
2. To understand the key concepts in data science, including their real-world applications and the toolkit used by data scientists.
3. To implement data collection and management scripts using MongoDB.
4. Critically evaluate data visualisations based on their design and use for communicating stories from data.
5. Evaluate outcomes and make decisions based on data
6. Effectively communicate results

TOPIC COVERED

UNIT-I **9**

Introduction to Data Science, Different Sectors using Data science, Purpose and Components of Python in Data Science. Applications of Data Science, Data Science and Ethical Issues- Discussions on privacy, security, ethics- A look back at Data Science- Next-generation data scientists.

UNIT-II **9**

Data Analytics Process, Knowledge Check, Exploratory Data Analysis (EDA), EDA- Quantitative technique, EDA- Graphical Technique, Data Analytics Conclusion and Predictions.

UNIT-III **9**

Feature Generation and Feature Selection (Extracting Meaning from Data)- Motivating application: user (customer) retention- Feature Generation (brainstorming, role of domain expertise, and place for imagination)- Feature Selection algorithms.

UNIT-IV **9**

Data Visualization- Basic principles, ideas and tools for data visualization, Examples of inspiring (industry) projects- Exercise: create your own visualization of a complex dataset.

TEXT BOOKS

1. Data Science from Scratch: First Principles with Python, Joel Grus, O'Reilly Media (2015).
2. An introduction to Data Science, Jeffrey Stanton.
3. Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython 2nd edition, William McKinney, O'Reilly Media (2017)
4. Business Analytics: The Science of Data - Driven Decision Making, U Dinesh Kumar, John Wiley & Sons.
5. Introducing Data Science: Big Data, Machine Learning, and More, Using Python Tools, Davy Cielen, John Wiley & Sons.
6. Data Science from Scratch, Joel Grus, O'Reilly Publisher Media.

REFERENCE BOOKS

1. Mining of Massive Datasets. v2.1, Jure Leskovek, Anand Rajaraman and Jeffrey Ullman. , Cambridge University Press.
2. Python Data Science Handbook, Jake VanderPlas, Shroff Publisher/O'Reilly Publisher Media.
3. Data Analysis with Open Source Tools, Philipp Janert, Shroff Publisher/O'Reilly Publisher Media.

ECS-104 INTRODUCTION TO ARTIFICIAL INTELLIGENCE

Course Category: PE1
Pre-requisite Subject: NIL
Contact Hours/Week: Lecture: 3, Tutorial: 1 & Practical: 0
Number of Credits: 4

Course Assessment methods: Continuous assessment through assignments, quizzes, tutorials and one minor test and one major examination.

Course Objective:
To review and strengthen important mathematical concepts required for AI & ML.
Introduce the concept of learning patterns from data and develop a strong theoretical foundation for understanding state of the art Machine Learning algorithms.

COURSE OUTCOMES:
1. Explain AI fundamentals and represent knowledge using predicate logic and rules.
2. Differentiate between procedural and declarative knowledge and apply logic programming basics.
3. Describe machine learning types and classify problems into regression and classification.
4. Implement and optimize linear regression using gradient descent.
5. Apply logistic regression for classification and handle multi-class and overfitting issues.
6. Understand and implement clustering algorithms with their real-world applications.

TOPIC COVERED

UNIT-I **9**
Defining Artificial Intelligence, Defining AI techniques, Using Predicate Logic and Representing Knowledge as Rules, Representing simple facts in logic, Computable functions and predicates, Procedural vs Declarative knowledge, Logic Programming, Mathematical foundations: Matrix Theory and Statistics for Machine Learning.

UNIT-II **9**
Idea of Machines learning from data, Classification of problem –Regression and Classification, Supervised and Unsupervised learning.

UNIT-III **9**

Linear Regression: Model representation for single variable, Single variable Cost Function, Gradient Decent for Linear Regression, Gradient Decent in practice.

UNIT-IV **9**

Logistic Regression: Classification, Hypothesis Representation, Decision Boundary, Cost function, Advanced Optimization, Multi-classification (One vs All), Problem of Overfitting.
Discussion on clustering algorithms and use-cases cantered around clustering and classification.

Text Books/References:

1. Saroj Kaushik, Artificial Intelligence, Cengage Learning, 1st Edition 2011.

2. Anindita Das Bhattacharjee, "Practical Workbook Artificial Intelligence and Soft Computing for beginners, Shroff Publisher-X team Publisher.

3. Yuxi (Hayden) Liu, "Python Machine Learning by Example", Packet Publishing Limited, 2017.

4. Tom Mitchell, Machine Learning, McGraw Hill, 2017.

5. Christopher M. Bishop, Pattern Recognition and Machine Learning, Springer, 2011.

6. T. Hastie, R. Tibshirani, J. Friedman. The Elements of Statistical Learning, 2e, 2011.

ECS-105 CRYPTOGRAPHY AND NETWORK SECURITY

Course Category: PE1
Pre-requisite Subject: NIL
Contact Hours/Week: Lecture: 3, Tutorial: 1 & Practical: 0
Number of Credits: 4

Course Assessment methods: Continuous assessment through assignments, quizzes, tutorials and one minor test and one major examination.

COURSE OUTCOMES:

After successful completion of the course, the learners would be able to:
1. Explain security attacks, services, mechanisms.
2. Describe modern block cipher designs including DES, AES.
3. Understand principles of public key cryptography, RSA, Diffie-Hellman.
4. Apply message authentication, hash functions, and analyze their security against attacks.
5. Explain digital signature algorithms, authentication protocols.
6. Describe IP security architecture, including authentication headers.

TOPIC COVERED

UNIT-I **9**

Introduction to security attacks - services and mechanism, Introduction to cryptography -
Conventional Encryption: Conventional encryption model, Classical encryption techniques -substitution ciphers and transposition ciphers – cryptanalysis, Steganography.

UNIT-II **9**

Stream and block ciphers - Modern Block Ciphers: Block ciphers principals - Shannon's theory of confusion and diffusion - Feistel structure - data encryption standard (DES) - strength of DES – differential and linear crypt analysis of DES - block cipher modes of operations - triple DES – AES.

UNIT-III **9**

Principles of public key crypto systems - RSA algorithm, Security of RSA - key management, Diffie-Hellman key exchange algorithm, Introductory idea of Elliptic curve cryptography, Elgamal encryption, Message Authentication and Hash Function, Authentication requirements & functions, message authentication code, birthday attacks, security of hash functions. MD5 message digest algorithm - Secure hash algorithm (SHA).

UNIT-IV **9**

Digital Signatures - authentication protocols - digital signature standards (DSS) - proof of digital signature algorithm - Authentication Applications: Kerberos and X.509 - directory authentication service - electronic mail security-pretty good privacy (PGP) - S/MIME, IP Security: Architecture - Authentication header - Encapsulating security payloads – combining security associations - key management.

Text Books:

1. William Stallings, "Crpyptography and Network security Principles and Practices", Pearson/PHI.
2. Wade Trappe, Lawrence C Washington, " Introduction to Cryptography with coding theory", Pearson.

Reference Books

1. W. Mao, "Modern Cryptography – Theory and Practice", Pearson Education.
2. Charles P. Pfleeger, Shari Lawrence Pfleeger – Security in computing – Prentice Hall of India.

REFERENCE BOOK

1. Sean Smith, "The Internet of Risky Things", Sean Smith, Shroff Publisher/O'Reilly Publisher.

ECS-106 COMPUTER GRAPHICS FOR VIRTUAL REALITY

Course Category: PE1
Pre-requisite Subject: NIL
Contact Hours/Week: Lecture: 3, Tutorial: 1 & Practical: 0
Number of Credits: 4

Course Assessment methods: Continuous assessment through assignments, quizzes, tutorials and one minor test and one major examination.

Course Objective:

1. To introduce the use of the components of a graphics system and become familiar with building approach of graphics system components and algorithms related with them.
2. To learn the basic principles of 3-dimensional computer graphics.
3. Provide an understanding of mapping from a world coordinates to device coordinates, clipping, and projections.

Course Outcomes: The students are expected to be able to demonstrate the following knowledge, skills, and attitudes after completing this course.

1. To list the basic concepts used in computer graphics.
2. To implement various algorithms to scan, convert the basic geometrical primitives, transformations, Area filling, clipping.
3. To define the fundamentals of animation, virtual reality and its related technologies.
4. To design an application with the principles of virtual reality.
5. To understand a typical graphics pipeline
6. To design an application with the principles of virtual reality

TOPIC COVERED

UNIT-I **9**

Graphics system and models: applications of computer graphics, graphics system, physical and synthetic images, imaging systems, graphics architectures, Geometric objects and transformations: scalars, points and vectors, three-dimensional primitives, coordinate systems and frames, frames in OpenGL, matrix and vector classes, modelling a colored cube, affine transformations - translation, rotation and scaling, transformations in homogeneous coordinates, concatenation of transformations, transformation matrices in OpenGL, interfaces to 3D applications, quaternion.

Vertices to fragments: basic implementation strategies, four major tasks, clipping - line clipping, polygon clipping, clipping of other primitives, clipping in three dimensions, polygon rasterization, hidden-surface removal, antialiasing, display considerations.

UNIT-II **9**

Lighting and shading: light and matter, light sources, the Phong reflection model, computation of vectors, polygonal shading, approximation of a sphere by recursive subdivision, specifying lighting parameters, implementing a lighting model, shading of the sphere model, per-fragment lighting, global illumination. Hierarchical modelling: symbols and instances, hierarchical models, a robot arm, trees and traversal, use of tree data structures, other tree structures, scene graphs, open scene graph.

UNIT-III **9**

Discrete techniques: buffers - digital images - writing into buffers - mapping methods - texture mapping - texture mapping in OpenGL - texture generation - environment maps - reflection map - bump mapping - compositing techniques - sampling and aliasing. Advanced rendering: going beyond pipeline rendering - ray tracing - building a simple ray tracer - the rendering equation - radiosity - Renderman - parallel rendering -volume rendering - Isosurfaces and marching cubes - mesh simplification - direct volume rendering - image-based rendering.

UNIT-IV **9**

Fractals: modelling - Sierpinski Gasket - coastline problem - fractal geometry - fractal dimension - recursively defined curves - Koch curves - c curves - dragons - space filling curves - turtle graphics - grammar based models - Graftals - volumetric examples - k-midpoint subdivision - fractal Brownian motion - fractal mountains - iteration in the complex plane - Mandelbrot set.

Virtual reality modelling language: introduction, exploring and building a world, building object, lighting, sound and complex shapes, animation and user interaction, colors, normals and textures, nodes references. Special applications: stereo display programming, multiport display systems, multi-screen display system, fly mode navigation, walk through navigation, virtual track ball navigation.

TEXT BOOKS

1. Rajesh K. Maurya, Computer Graphics with Virtual Reality System, John Wiley & Sons.
2. Edward Angel, "Interactive Computer Graphics: A Top-Down Approach Using OpenGL", Addison-Wesley.
3. Foley James D, Van Dam, Feiner and Hughes, "Computer Graphics: Principles and Practice", Pearson Education.

REFERENCE BOOK

1. Computer Graphics with Virtual Reality System, 3Ed, Rajesh K. Maurya, Wiley India

ECS-107 FUNDAMENTAL OF BLOCKCHAIN TECHNOLOGY AND APPLICATIONS

Course Category: PE1
Pre-requisite Subject: NIL
Contact Hours/Week: Lecture: 3, Tutorial: 1 & Practical: 0
Number of Credits: 4

Course Assessment methods: Continuous assessment through assignments, quizzes, tutorials and one minor test and one major examination.

Course Objective: It contains a basic introduction to familiarize students with the basics of advanced programming techniques. Here are the key objectives covered in this course.

1. To understand the concepts of blockchain technology
2. To understand the consensus and hyper ledger fabric in blockchain technology
3. Analyze the architecture and consensus mechanisms of Bitcoin.
4. Evaluate smart contract languages in automating and enforcing digital agreements.
5. Explore privacy-enhancing technologies in blockchain vulnerabilities.

Course Outcome: The students are expected to be able to demonstrate the following knowledge, skills, and attitudes after completing this course.

1. State the basic concepts of block chain
2. Paraphrase the list of consensus and Demonstrate and Interpret working of Hyper ledger Fabric

3. Implement SDK composer tool and explain the Digital identity for government
4. Compare and contrast different blockchain platforms.
5. Identify appropriate blockchain platforms for specific use cases.
6. Testing and debugging blockchain applications

TOPIC COVERED

UNIT-I **9**

History: Digital Money to Distributed Ledgers -Design Primitives: Protocols, Security, Consensus, Permissions, Privacy: Block chain Architecture and Design-Basic crypto primitives: Hash, Signature, Hash chain to Block chain-Basic consensus mechanisms.

UNIT-II **9**

Requirements for the consensus protocols-Proof of Work (PoW)-Scalability aspects of Block chain consensus protocols: Permissioned Block chains-Design goals-Consensus protocols for Permissioned Block chains, Zcash and Zk-SNARKS for anonymity preservation, attacks on Blockchains – such as Sybil attacks, selfish mining, 51% attacks.

UNIT-III **9**

Decomposing the consensus process-Hyper ledger fabric components-Chain code Design and Implementation: Hyper ledger Fabric II:-Beyond Chain code: fabric SDK and Front End-Hyper ledger composer tool, Using smart contracts to enforce legal contracts, comparing Bitcoin scripting vs. Ethereum Smart Contracts

UNIT-IV **9**

Block chain in Financial Software and Systems (FSS): -Settlements, -KYC, -Capital markets-InsuranceBlock chain in trade/supply chain: Provenance of goods, visibility, trade/supply chain finance, invoice management/discounting. Blockchain for Government: Digital identity, land records and other kinds of record keeping between government entities, public distribution system / social welfare systems: Block chain Cryptography: Privacy and Security on Block chain.

Textbooks:

1. Mark Gates, “Block chain: Ultimate guide to understanding block chain, bit coin, crypto currencies, smart contracts and the future of money”, Wise Fox Publishing and Mark Gates 2017.
2. Salman Baset, Luc Desrosiers, Nitin Gaur, Petr Novotny, Anthony O'Dowd, Venkatraman Ramakrishna, “Hands-On Block chain with Hyper ledger: Building decentralized applications with Hyperledger Fabric and Composer”, 2018.
3. Bahga, Vijay Madisetti, “Block chain Applications: A Hands-On Approach”, ArshdeepBahga, Vijay Madisetti publishers 2017.

ECS-108 INTRODUCTION TO C PROGRAMMING

Course Category: PE1
Pre-requisite Subject: NIL

Contact Hours/Week: Lecture: 3, Tutorial: 0 & Practical: 2
Number of Credits: 4

Course Assessment methods: Continuous assessment through assignments, quizzes, practical work, record, viva voce and one minor test and one major examination.

Course Objective: The course covers the basics of programming and demonstrates fundamental programming techniques, customs and terms including the most common library functions and the usage of the pre-processor.

1. To develop C Programs using basic programming constructs
2. To develop C programs using arrays and strings
3. To develop applications in C using functions and structures

Course Outcomes: The students are expected to be able to demonstrate the following knowledge, skills, and attitudes after completing this course.

1. Basic terminology used in computer programming.
2. Programs development in C Language by writing, compiling and debugging.
3. Design of programs involving simple statements, conditional statements, iterative statements, array, strings, functions, recursion, structure and union.
4. Difference between call by value and call by reference and dynamic memory allocations and use of pointers.
5. Basic operations on a file.
6. Basics of dynamic memory.

TOPIC COVERED

UNIT-I **9**

Fundamentals of C Programming: Structure of C Program, Writing and Executing the First C Program, Components of C Language, Standard I/O, Formatted I/O. Conditional Program Execution: Applying if and switch Statements, Nesting if and else. Program Loops and Iterations: Use of while, do while and for Loops, Multiple Loop Variables, Use of break and continue Statements, goto Statement.

UNIT-II **9**

Arrays: One Dimensional, Multidimensional Array and Their Applications, Declaration and Manipulation of Arrays. Strings: String Variable, String Handling Functions, Array of Strings.
Functions: Designing Structured Programs, Functions in C, User Defined and Standard Functions, Formal vs. Actual Arguments, Function Category, Function Prototype, Parameter Passing, Recursive Functions. Storage Classes revisited.

UNIT-III **9**

Pointers: Pointer Variable and its Importance, Pointer Arithmetic Pointers and Arrays, Pointer and Character Strings, Pointers and Functions, Array of Pointers, Pointers to Pointers. Structure: Declaration

and Initialization of Structures, Structure as Function Parameters, Structure Pointers. Union: Declaration and Initialization of Unions, Union as Function Parameters, Union Pointers.

UNIT-IV **9**

Dynamic Memory Allocation: malloc, calloc, realloc, free functions. File Management: Defining and Opening a File, Closing a File, Input/ Output Operations in Files. The Pre-processor Directives, Macros. Command Line Arguments. Introduction to Graphics Programming.

Textbooks

1. Jeri R. Hanly and Elliot B. Koffman, Problem Solving and Program Design in C, 7th Edition, Pearson.
2. Schildt, Herbert, Complete Reference with C, Tata McGraw Hill.
3. Kerninghan and Ritchie, The C programming Language, 2nd Edition, Prentice Hall.
4. Richard Bird, Introduction to Functional Programming using Haskell, 2nd Edition, Prentice- Hall International, 1998.

Reference Books

1. Greg Michaelson, An Introduction to Functional Programming Through Lambda Calculus, Dover Edition, Addition Wesley Publication.
2. Samuel P. Harbison, and Guy L. Steele Jr., C-A Reference Manual, Fifth Edition, Prentice Hall, 2002.

List of Experiments

Write a program to print “Hello, World!” and accept user input using formatted I/O.

1. Implement a program to find the largest of three numbers using nested if-else statements.
2. Create a program that prints all prime numbers between 1 and N using for, while, and do-while loops.
3. Write a program to reverse a string using arrays and string handling functions.
4. Develop a program to implement a recursive function for calculating factorial of a number.
5. Write a program to add two matrices using multidimensional arrays.
6. Create a program to swap two numbers using pointers and demonstrate pointer arithmetic.
7. Design a program to store student information using structures and print the details.
8. Write a program to dynamically allocate memory for an array and perform insertion and deletion of elements.
9. Implement file operations: write data to a file and read it back to display on the screen.
10. Use command line arguments to accept input parameters and print them.
11. Create simple macros using preprocessor directives to find the maximum and minimum of two numbers.

ECS-109 GLOBAL POSITIONING SYSTEM

Course Category: PE1
Pre-requisite Subject: NIL

Contact Hours/Week: Lecture: 3, Tutorial: 1 & Practical: 0
Number of Credits: 4

Course Assessment methods: Continuous assessment through assignments, quizzes, tutorials and one minor test and one major examination.

Course Outcomes:

1. Explain the fundamental concepts of geodesy, Earth's shape, gravity field, and atmospheric influence on geodetic measurements.
2. Describe the architecture of GPS, including satellite constellation, signal structure, coordinate systems (WGS-84), and timing systems.
3. Identify and analyze GPS signal errors and biases affecting positioning accuracy.
4. Apply GPS orbital and navigational principles to determine precise positions using field observations and GPS receivers.
5. Process GPS data and integrate it with GIS platforms for spatial analysis and mapping applications.

Topics Covered

Unit I **9**

Introduction to geodesy and its development , Earth and its size and shape, earth and its motions- annual, spin, precession, nutation, polar motion, Earth and its gravity field – anomaly, gravity potential, geoid and deflection to vertical,. Earth and its atmosphere – physical properties, wave propagation through atmosphere, temporal variations, gravitational field of the atmosphere,

Unit II **9**

Introduction of Global Positioning System, Satellite constellation, GPS signals and data, pseudo range and carrier phase measurements; Signal structure; GPS coordinate systems: WGS-84, GPS time; GPS Errors and biases.

Unit III **9**

GPS orbital Geometry and Navigational solution; GPS receiver, GPS antenna. Radio and its types, Radio Antenna, Surveying with GPS; Planning and field observations;

Unit IV **9**

Data post-processing; GIS and GPS integration; Other satellite based navigational systems: GLONASS, GALILEO, modernization plans of navigational satellites. GPS Applications

Text &Reference Books:

1. Understanding GPS/GNSS: Principles and Applications (Authors: Elliott D. Kaplan and Christopher Hegarty), 3rd Edition, 2011, Publisher: Artech House
2. Gps Satellite Surveying, by Alfred Leick, Repoport, Tatarikov, 4th Edition, Wiley Publishing

House.

3. Global Positioning System: Theory and Practice (Authors: B. Hofmann-Wellenhof, H. Lichtenegger, J. Collins), 5th Edition, 2001, Publisher: Springer.
4. Global Positioning System: Theory and Applications, by Branford W. Parkinson.

ECS-201 ADVANCED PROGRAMMING TECHNIQUES

Course Category: PE2
Pre-requisite Subject: NIL
Contact Hours/Week: Lecture: 3, Tutorial: 1/0 & Practical: 0/2
Number of Credits: 4

Course Assessment Methods: Continuous Assessment through One Minor Test, Teacher Assessment (Quiz, Tutorial, Assignment, Attendance), Practical Work & Viva-voce, and One Major Theory & One Practical Examinations.

Course Objective: It contains a basic introduction to familiarize students with the basics of advanced programming techniques. Here are the key objectives covered in this course.

1. The focus in this course is on problem solving within the object-oriented programming paradigm
2. Read and understand software specifications to implement code that conforms to the specifications and to course coding standards.
3. Use advanced programming techniques to solve computing problems.
4. Proficiently use fundamental programming elements including: variable declaration, use of data types and simple data structures (arrays and objects), decision structures, loop structures, input and output for console and text files, and functions/methods

Course Outcomes: The students are expected to be able to demonstrate the following knowledge, skills, and attitudes after completing this course.

1. Develop algorithms from user problem statements.
2. Express the solutions to computer oriented problems using pseudocode.
3. Proficiently transform designs of problem solutions into a standard programming language.
4. Use an integrated programming environment to write, compile, and execute programs involving a small number of source files.
5. Apply debugging and testing techniques to locate and resolve errors, and to determine the effectiveness of a program.
6. Apply standard/structured programming techniques including design approaches, use of functions/methods, use of documentation, and avoidance of excessive branching.

TOPIC COVERED

UNIT-I **9**

Introduction-History of Computers, Components of a Computer, Programming Languages, Compilation vs. Interpretation, Basic Program Structure and the Integrated Development Environment-Essential Program Structure, Documentation and Standard Programming Practices, Integrated Development Environment (IDE) Overview, Editing (with the IDE), Compilation (with the IDE), Execution (with the IDE), Debugging (with the IDE)

UNIT-II **9**

Algorithm Development using Psuedo-code-Software Engineering Method, Procedural Problem Solving Approaches, Assignments, Conditionals, Loops, Classic Formula Problems, Classic Aggregate Problems (E.G., Maximum, Minimum, Sum, Average), Basic Input and Output-Console Output including Basic Data Formatting, Console Input Variables and Expressions-Variable Declarations including Common Data Types (E.G. Int, Float, String), Arithmetic, Expressions Including Precedence and Associativity, Assignment Statements (Numeric and String Data), Library Functions, Standard Programming Practices for Variables and Assignments, Case Problems Using Variables and Expressions

UNIT-III **9**

Decision Structures-Boolean Expressions, Single Alternative Conditional Statements (E.G., If), Double Alternative Conditional Statements (E.G., If/Else), Multi-Way Statements (E.G., Case), Nested Conditional Structures, Standard/ Structures Programming Practices for Decision Structures, Case Problems using Decisions Structures

Loop Structures-Loop Control Variables, Initialization, Test and Modifications, Pre-Test Loop (E.G., While Loop), Post-Test Loop (E.G., Do-While Loop), Counting Loop (E.G., For Loop), Nested Loop Structures, Standard/ Structures Programming Practice for Loop Structures, Case Problems using Loop Structures

Input and Output using Files-Input Streams from Files, Priming Read Loop, Output Streams to Files, Case Problems using File Input and Output

UNIT-IV **9**

Simple Data Structures-One Dimensional Arrays, Strings as Arrays, Multi-Dimensional Arrays, Records (E.G., Objects/Entities), Case Problems using Arrays and Records

Functions-Argument Passing, Returning Results, Recursion, Testing A Program System, Standard/Structures Programming Practices for Functions, Case Problems using Functions

Introduction to the Object-Oriented Approach-Class Declarations, Instance Variables, Methods, Object Instantiation, Standard/Structures Programming Practice for Classes, Case Problems using Objects

Textbooks:

1. Gaddis Tony, Starting Out with C++: From control structures through objects, 7th Edition, Addison-Wesley Publishing, 2012.
2. Lewis, John, and Loftus, William, JAVA Software Solutions: Foundations of Program Design, 7th Edition, Pearson, 2012.
3. Stroustrup, Bjarne, Programming: Principles and Practice Using C++, Addison-Wesley Professional, 2008

ECS-202 FUNDAMENTAL OF BLOCKCHAIN

Course Category: PE2
Pre-requisite Subject: NIL
Contact Hours/Week: Lecture: 3, Tutorial: 1/0 & Practical: 0/2
Number of Credits: 4

Course Assessment Methods: Continuous Assessment through One Minor Test, Teacher Assessment (Quiz, Tutorial, Assignment, Attendance), Practical Work & Viva-voce, and One Major Theory & One Practical Examinations.

Course Objective: It contains a basic introduction to familiarize students with the basics of advanced programming techniques. Here are the key objectives covered in this course.

6. To understand the concepts of block chain technology
7. To understand the consensus and hyper ledger fabric in block chain technology

Course Outcomes: The students are expected to be able to demonstrate the following knowledge, skills, and attitudes after completing this course.

7. State the basic concepts of block chain
8. Paraphrase the list of consensus and Demonstrate and Interpret working of Hyper ledger Fabric
9. Implement SDK composer tool and explain the Digital identity for government
10. Compare and contrast different blockchain platforms.
11. Identify appropriate blockchain platforms for specific use cases.
12. Testing and debugging blockchain applications

TOPIC COVERED

UNIT-I **9**

History: Digital Money to Distributed Ledgers -Design Primitives: Protocols, Security, Consensus, Permissions,
Privacy: Block chain Architecture and Design-Basic crypto primitives: Hash, Signature, Hash chain to Block chain-Basic consensus mechanisms.

UNIT-II **9**

Requirements for the consensus protocols-Proof of Work (PoW)-Scalability aspects of Block chain consensus protocols: Permissioned Block chains-Design goals-Consensus protocols for Permissioned Block chains.

UNIT-III **9**

Decomposing the consensus process-Hyper ledger fabric components-Chain code Design and Implementation: Hyper ledger Fabric II:-Beyond Chain code: fabric SDK and Front End-Hyper ledger composer tool.

UNIT-IV **9**

Block chain in Financial Software and Systems (FSS): -Settlements, -KYC, -Capital markets-InsuranceBlock chain in trade/supply chain: Provenance of goods, visibility, trade/supply chain finance, invoice management/discounting. Block chain for Government: Digital identity, land records and other kinds of record keeping between government entities, public distribution system / social welfare systems: Block chain Cryptography: Privacy and Security on Block chain.

Textbooks:

4. Mark Gates, "Block chain: Ultimate guide to understanding block chain, bit coin, crypto currencies, smart contracts and the future of money", Wise Fox Publishing and Mark Gates 2017.
5. Salman Baset, Luc Desrosiers, Nitin Gaur, Petr Novotny, Anthony O'Dowd, Venkatraman Ramakrishna, "Hands-On Block chain with Hyper ledger: Building decentralized applications with Hyperledger Fabric and Composer", 2018.
6. Bahga, Vijay Madisetti, "Block chain Applications: A Hands-On Approach", Arshdeep Bahga, Vijay Madisetti publishers 2017.

ECS-203 SOFTWARE ENGINEERING

Course Category: PE2
Pre-requisite Subject: NIL
Contact Hours/Week: Lecture: 3, Tutorial: 1/0 & Practical: 0/2
Number of Credits: 4

Course Assessment Methods: Continuous Assessment through One Minor Test, Teacher Assessment (Quiz, Tutorial, Assignment, Attendance), Practical Work & Viva-voce, and One Major Theory & One Practical Examinations.

Course Objective: It contains a basic introduction to familiarize students with the basics of software Engineering. Here are the key objectives covered in this course

1. The aim of the course is to provide an understanding of the working knowledge of the techniques for estimation, design, testing and quality management of large software development projects.
2. Topics include process models, software requirements, software design, software testing, software process/product metrics, risk management, quality management and UML diagrams

Course Outcomes: The students are expected to be able to demonstrate the following knowledge, skills, and attitudes after completing this course.

1. List and describe the fundamental phases of the Software Development Life cycle (SDLC)
2. Define and describe fundamental software engineering terminology and coding practices
3. Explore/explain relationships between software engineering and other engineering disciplines (Systems Engineering, Electrical and Computer Engineering, Industrial Engineering)
4. Modify/build a software program that introduces students to software development tools / environments
5. Troubleshoot and debug changes made to an existing software program
6. Build a foundation for academic success in the Software Engineering degree program

TOPIC COVERED

UNIT-I **9**

Software Process – Introduction, S/W Engineering Paradigm, Life Cycle Models (Waterfall, Incremental, Spiral, Evolutionary, Prototyping), Software Requirements –Functional And Non-Functional – Software Document – Requirement Engineering Process – Feasibility Studies – Software Prototyping – Prototyping in Software, Process – Data – Functional and Behavioral Models – Structured Analysis And Data Dictionary.

UNIT-II **9**

Basic Concept of Software Design, Architectural Design, Low Level Design: Modularization, Design Structure Charts, Pseudo Codes, Flow Charts, Coupling and Cohesion Measures, Design Strategies: Function Oriented Design, Object Oriented Design, Top-Down and Bottom-Up Design. Software Measurement and Metrics: Various Size Oriented Measures: Halestead's Software Science, Function Point (FP) Based Measures, Cyclomatic Complexity Measures: Control Flow Graph.

UNIT-III **9**

Software Testing – Taxonomy of S/W Testing Levels - Black Box Testing – Testing Boundary Conditions – Structural Testing — Regression Testing– S/W Testing Strategies, Unit Testing, Integration Testing, Validation Testing, System Testing and Debugging.

UNIT-IV **9**

Measures and Measurements – Zipf's Law, Software Cost Estimation – Function Point Models, COCOMO Model. Delphi Method – Scheduling – Earned Value Analysis – Error Tracking – Software Configuration Management – Program Evolution Dynamics – Software Maintenance – Project Planning – Project Scheduling– Risk Management – Case Tools

TEXT BOOKS

1. R. S. Pressman, "Software Engineering - A practitioners approach", 3rd Edition, McGraw Hill International editions, 1992.
2. IAN Sommerville, Software Engineering, Pearson Education Asia, VI Edition, 2000
3. Pankaj Jalote, "An Integrated Approach to software Engineering", Springer Verlag, 1997

SKILL-BASED COURSES TO QUALIFY FOR UG DIPLOMA (ENGG.) IN COMPUTER SCIENCE & ENGINEERING

BCS-264 **DATA ANALYTICS USING PYTHON**

Course Category: Skill Enhancement
Pre-requisite Subject: NIL
Contact Hours/Week: Lecture: 2, Tutorial: 0 & Practical: 2
Number of Credits: 3

Course Assessment Methods: Continuous assessment through assignments, quizzes, tutorials, examination, and lab-based project evaluations.

Course Objective:

1. To provide knowledge and expertise to perform data analytics using Python and its libraries.
2. To demonstrate an understanding of data analytics concepts, including data collection, processing, and visualization.
3. To produce Python code to analyze datasets and derive actionable insights.

Course Outcomes: The students are expected to be able to demonstrate the following knowledge, skills, and attitudes after completing this course:

1. Explain the data analytics pipeline and the role of Python in data analysis.
2. Understand key concepts in data analytics, including real-world applications and tools used by data analysts.
3. Implement data processing and analysis scripts using Python libraries like Pandas and NumPy.
4. Create effective data visualizations to communicate insights using Matplotlib and Seaborn.
5. Evaluate and interpret analytical results to support data-driven decision-making.
6. Effectively present analytical findings through reports and visualizations.

Topics Covered

Unit-I: Introduction to Data Analytics and Python 6
Overview of data analytics: Definition, importance, and applications; Python for data analytics: features, libraries (Pandas, NumPy, Matplotlib); data types and sources (CSV, JSON, databases); setting up Python environment (Jupyter Notebook, VS Code); ethical considerations: data privacy, bias, and security; case studies: real-world data analytics projects.

Unit-II: Data Collection and Preprocessing 6
Data collection methods: APIs, web scraping, and databases; data cleaning: handling missing values, duplicates, and outliers; data transformation: normalization, encoding, and scaling; exploratory data analysis (EDA): summary statistics and correlations; Python libraries: Pandas for data manipulation, NumPy for numerical operations; exercise: preprocess a dataset using Pandas.

Unit-III: Data Analysis and Statistical Techniques 6
Statistical foundations: mean, median, variance, and distributions; data analysis techniques: grouping, filtering, and aggregation; feature engineering: creating and selecting relevant features; introduction to machine learning: regression and classification basics; Python libraries: SciPy, Scikit-learn; exercise: analyze a dataset to identify trends and patterns.

Unit-IV: Data Visualization and Reporting 6
Data visualization principles: design, clarity, and storytelling; visualization tools: Matplotlib, Seaborn, and Plotly; creating charts: bar, line, scatter, and heatmaps; reporting insights: dashboards and summary reports; sharing results: exporting visualizations and reports; project: develop a data analytics project with visualizations and insights (e.g., sales or social media analysis).

Text and Reference Books

1. *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython*, Wes McKinney, O'Reilly Media (2017).
2. *Data Science for Beginners: 4 Books in 1*, Andrew Park, Independently Published (2020).
3. *Practical Data Science with Python*, Nathan George, Packt Publishing (2021).
4. *Python Data Science Handbook*, Jake VanderPlas, O'Reilly Media (2016).

List of Experiments

1. Set up a Python data analytics environment with Jupyter Notebook and install Pandas, NumPy, and Matplotlib.
2. Collect and preprocess a CSV dataset (e.g., sales data) by handling missing values and outliers using Pandas.
3. Perform exploratory data analysis (EDA) on a dataset to compute summary statistics and correlations.
4. Create visualizations (bar, line, and scatter plots) for a dataset using Matplotlib and Seaborn.
5. Build a simple linear regression model using Scikit-learn to predict trends in a dataset.
6. Develop a dashboard with multiple visualizations to summarize insights from a dataset (e.g., customer data).
7. Complete a mini-project: Analyze a public dataset (e.g., COVID-19 or e-commerce) and present findings with visualizations.

BCS-265 MOBILE APPLICATION DEVELOPMENT

Course Category: Skill Enhancement
Pre-requisite Subject: NIL
Contact Hours/Week: Lecture: 2, Tutorial: 0 & Practical: 2
Number of Credits: 3

Course Assessment Methods: Continuous assessment through assignments, quizzes, tutorials, one minor test, one major examination, and lab-based project evaluations.

Course Objective:

1. To provide knowledge and expertise to develop cross-platform mobile applications using Flutter and Dart.
2. To demonstrate an understanding of mobile app development concepts, including UI design, state management, and API integration.
3. To produce Flutter code to create functional and responsive mobile applications.

Course Outcomes: The students are expected to be able to demonstrate the following knowledge, skills, and attitudes after completing this course:

1. Explain the architecture and components of mobile applications and the Flutter framework.
2. Understand key concepts in mobile app development, including their real-world applications and tools used by developers.
3. Implement UI designs and navigation using Flutter widgets and Dart programming.
4. Integrate APIs and external data sources into mobile applications.
5. Evaluate and debug mobile applications to ensure functionality and performance.

6. Effectively deploy and present mobile apps for Android and iOS platforms.

Topics Covered

Unit-I: Introduction to Mobile Application Development 6
Overview of native vs. cross-platform frameworks; Flutter and Dart features, architecture, setup; Android/iOS ecosystem, app stores; IDE setup (VS Code/Android Studio), emulators, SDKs; ethical issues (privacy, permissions, security); case studies of Flutter apps.

Unit-II: User Interface Design with Flutter 6
Flutter widgets (stateless, stateful, widget tree); responsive layouts (Flexbox, rows, columns, stacks); UI components (buttons, text fields, images, navigation); styling (themes, fonts, colors); accessibility best practices; exercise: design a simple app UI.

Unit-III: State Management and App Functionality 6
State management (setState, Provider, Riverpod); navigation (routes, named routes); user input (forms, validation); local storage (SharedPreferences, SQLite); debugging/testing (unit, widget tests); exercise: build a to-do list app with state and storage.

Unit-IV: API Integration and Deployment 6
REST APIs, HTTP requests, JSON parsing; external services (Firebase, third-party libraries); app optimization, error handling; deployment (APKs for Android, IPA for iOS); app store guidelines; project: develop and deploy a functional app (e.g., weather, e-commerce).

Text and Reference Books

1. Flutter for Beginners: An introductory guide to building cross-platform mobile applications, Alessandro Biessek, Packt Publishing (2020).
2. Beginning App Development with Flutter: Create Cross-Platform Mobile Apps, Rap Payne, Apress (2019).
3. Dart Apprentice, Jonathan Sande & Matt Galloway, Razeware LLC (2021).
4. Flutter in Action, Eric Windmill, Manning Publications (2020).
5. Google Flutter Mobile Development Quick Start Guide, Salvatore Giordano, Packt Publishing (2019).

List of Experiments

Set up a Flutter development environment and run a sample app on an emulator.

1. Create a responsive UI for a login screen using Flutter widgets.
2. Build a to-do list app with state management using Provider or Riverpod.
3. Develop a weather app that fetches data from a public API (e.g., OpenWeatherMap).
4. Implement local storage in an app using SQLite to save user preferences.
5. Debug a Flutter app using Dart DevTools to identify and fix performance issues.
6. Build and deploy a simple app (e.g., a note-taking app) to an Android emulator or device.