Bharati Vidyapeeth

(Deemed to be University)

Pune, India

College of Engineering, Pune

B.Tech. (Computer Engineering) Program Curriculum

(2021 Course)

VISION OF UNIVERSITY:

Social Transformation Through Dynamic Education.

MISSION OF UNIVERSITY:

- To make available quality education in different areas of knowledge to the students as per their choice and inclination
- To offer education to the students in a conducive ambience created by enriched infrastructure! and academic facilities in its campuses.
- To bring education within the reach of rural, tribal and girl students by providing them substantive fee concessions and subsidized hostel and mess facilities
- To make available quality education to the students of rural, tribal, and other deprived sections of the population

VISION OF THE INSTITUTE:

To be World Class Institute for Social Transformation Through Dynamic Education.

MISSION OF THE INSTITUTE:

- To provide quality technical education with advanced equipment, qualified faculty members, infrastructure to meet needs of profession and society.
- To provide an environment conducive to innovation, creativity, research, and entrepreneurial leadership.
- To practice and promote professional ethics, transparency and accountability for social community, economic and environmental conditions.

VISION OF THE DEPARTMENT

To pursue and excel in the endeavour for creating globally recognised Computer Engineers through Quality education.

MISSION OF THE DEPARTMENT

- To impart engineering knowledge and skills confirming to a dynamic curriculum.
- To develop professional, entrepreneurial & research competencies encompassing continuous intellectual growth.
- To produce qualified graduates exhibiting societal and ethical responsibilities in working environment.

PROGRAM EDUCATIONAL OBJECTIVES

The students of B.TECH. (COMPUTER ENGINEERING), after graduating with Bachelor of Technology degree in Computer Engineering, will able to

1. Demonstrate technical and professional competencies by applying engineering fundamentals, computing principles and technologies.
2. Learn, Practice, and grow as skilled professionals/entrepreneur/researchers adapting to the evolving computing landscape.
3. Demonstrate professional attitude, ethics, understanding of social context and interpersonal skills leading to a successful career.

PROGRAM SPECIFIC OUTCOMES

1. To design, develop and implement computer programs on hardware towards solving problems.
2. To employ expertise and ethical practise through continuing intellectual growth and adapting to the working environment.

PROGRAM OUTCOMES

a. Apply the knowledge of mathematics, science, engineering fundamentals, and computing for the solution of complex engineering problems.
b. Identify, formulate, research literature, and analyse complex engineering problems reaching substantiated conclusions using computer engineering foundations, principles, and technologies.
c. Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for public health and safety, and cultural, societal, and environmental considerations.
d. Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
e. Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.
f. Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal, and cultural issues, and the consequent responsibilities relevant to the professional engineering practice.
g. Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and the need for sustainable development.
h. Apply ethical principles while committed to professional responsibilities and norms of the engineering practice.
i. Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings
j. Communicate effectively on complex engineering activities with the engineering community and with the society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
k. Apply the engineering and management principles to one's work, as a member and leader in a team.
l. Recognise the need for and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

CORELATION BETWEEN GRADUATE ATTRIBUTES AND PROGRAMME OUTCOMES

Graduate Attributes/ Programme Outcomes	a	b	c	d	e	f	g	h	i	j	k	l
Engineering Knowledge	✓											
Problem Analysis		✓										
Design/Development of Solutions			✓									
Conduct Investigations of Complex Problems				✓								
Modern Tool Usage					✓							
The Engineer and Society						✓						
Environment and Sustainability							✓					
Ethics								✓				
Individual and Teamwork									✓			
Communication										✓		
Project Management and Finance											✓	
Life-Long Learning												✓

DEFINITION OF CREDITS:

1 Hour Lecture (L) per week	1 credit
1 Hour Tutorial (T) per week	1 credit
2 Hour Practical (P) per week	1 credit
4 Hours Practical (P) per week	2 credits

STRUCTURE OF UNDERGRADUATE ENGINEERING PROGRAMME

Sr. No.	Category	Breakup of Credits
1	Basic Science Course (BSC)	26
2	Engineering Science Course (ESC)	10
3	Core Courses (CC)	143
4	Elective Courses (EC)	10
5	Project (PROJ)	18
6	Internship (INT)	03
7	Vocational Course (VC)	08
8	Massive Open Online Course (MOOC)	06
9	Research Paper Publication (Research)	02
10	Social Activities (SA)	04
11	Mandatory Course (MC)	Non-Credit
12	Internal Assessment (IA)	-
13	University Examination (UE)	-
TOTAL		**230**

DISTRIBUTION OF COURSE COMPONENTS

Sr. No.	Category	Number of Courses
1	Basic Science Courses (BSC)	07
2	Engineering Science Course (ESC)	02
3	Core Courses (CC)	36
4	Elective Courses (EC)	02
5	Project (PROJ)	04
6	Internship (INT)	01
7	Vocational Courses (VC)	04
8	Massive Open Online Courses (MOOC)	03
9	Research Paper Publication (Research)	01
10	Social Activities (SA)	02
11	Mandatory Courses (MC)	01
TOTAL		**63**

Program: B.TECH. (Computer Engineering) Semester – I 2021 Course

Sr. No.	Name of Course	Teaching Scheme (Hrs./Week)			Examination Schemes (Marks)						Credits			
		L	P	T	UE	IA	TW	TW & OR	TW & PR	Total	L	P (TW/PR/OR)	T	Total
1	Mathematics for Computing - I	3	-	1	60	40	-	-	-	100	3	-	1	4
2	Organic and Electrochemistry	3	2	-	60	40	25	-	-	125	3	1	-	4
3	Digital Electronics	4	2	-	60	40	25	-	-	125	4	1	-	5
4	Classic Data Structures	4	2	-	60	40	-	-	50	150	4	1	-	5
5	Computational Thinking and Programming Concepts	4	2	-	60	40	-	-	100	200	4	1	-	5
6	Programming Technologies and Tools Laboratory– I	-	4	-	-	-	-	-	50	50	-	2	-	2
		18	**12**	**1**	**300**	**200**	**50**	**-**	**200**	**750**	**18**	**6**	**1**	**25**

Program: B.TECH. (Computer Engineering) Semester – II 2021 Course

Sr. No.	Name of Course	Teaching Scheme (Hrs./Week)			Examination Schemes (Marks)						Credits			
		L	P	T	UE	IA	TW	TW & OR	TW & PR	Total	L	P (TW/PR/OR)	T	Total
1	Mathematics for Computing - II	3	-	1	60	40	-	-	-	100	3	-	1	4
2	Physics for Computing Systems	3	2	-	60	40	25	-	-	125	3	1	-	4
3	Numerical Computation	4	-	-	60	40	-	-	-	100	4	-	-	4
4	Electrical Technology	4	2	-	60	40	25	-	-	125	4	1	-	5
5	Paradigms of Programming	4	2	-	60	40	-	-	50	150	4	1	-	5
6	Programming Technologies and Tools Laboratory - II	-	4	-	-	-	-	-	50	50	-	2	-	2
7	Computer System Workshop Technology	-	2	-	-	-	-	-	100	100	-	1	-	1
		18	**12**	**1**	**300**	**200**	**50**	**-**	**200**	**750**	**18**	**6**	**1**	**25**

Program: B.TECH. (Computer Engineering) Semester – III 2021 Course

Sr. No.	Name of Course	Teaching Scheme (Hrs./Week)			Examination Schemes (Marks)						Credits			
		L	P	T	UE	IA	TW	TW & OR	TW & PR	Total	L	P (TW/PR/OR)	T	Total
1	Discrete Mathematics and Applications	3	-	1	60	40	-	-	-	100	3	-	1	4
2	Data Structures and Algorithmic Thinking	4	2	-	60	40	-	-	75	175	4	1	-	5
3	Computer Organisation and Design	4	2	-	60	40	-	-	50	150	4	1	-	5
4	Computer Networks	3	2	-	60	40	-	-	75	175	3	1	-	4
5	Software Engineering*	4	4	-	60	40	-	50	-	150	4	2	-	6
6	Programming Technologies and Tools Laboratory – III	-	4	-	-	-	-	-	50	50	-	2	-	2
7	Vocational Course - I	-	-	-	-	-	-	-	-	-	-	-	-	2
8	MOOC-I	-	-	-	-	-	-	-	-	-	-	-	-	2
9	Environmental Studies** (Mandatory Audit Course)	-	-	-	-	-	-	-	-	-	-	-	-	-
		18	**14**	**1**	**300**	**200**	**-**	**50**	**250**	**800**	**18**	**7**	**1**	**30**

***Industry Taught Course – I** ** 50 Marks Theory Examination**

List of MOOCs and Vocational Courses will be published by the department before the commencement of respective semester.

Program: B.TECH. (Computer Engineering) Semester – IV 2021 Course

Sr. No.	Name of Course	Teaching Scheme (Hrs./Week)			Examination Schemes (Marks)						Credits			
		L	P	T	UE	IA	TW	TW & OR	TW & PR	Total	L	P (TW/PR/OR)	T	Total
1	Probability and Statistics	3	-	-	60	40	-	-	-	100	3	-	-	3
2	Models of Computation	3	-	1	60	40	-	-	-	100	3	-	1	4
3	Computer Operating System	3	2	-	60	40	-	-	50	150	3	1	-	4
4	Database Management System	3	2	-	60	40	-	-	50	150	3	1	-	4
5	Wireless Communication*	4	2	-	60	40	-	50	-	150	4	1	-	5
6	Design Thinking and Communication	2	4	-	-	-	-	-	100	100	2	2	-	4
7	Programming Technologies and Tools Laboratory – IV	-	4	-	-	-	-	-	50	50	-	2	-	2
8	Vocational Course- II	-	-	-	-	-	-	-	-	-	-	-	-	2
9	Social Activities - I	-	-	-	-	-	-	-	-	-	-	-	-	2
		18	**14**	**1**	**300**	**200**	**-**	**50**	**250**	**800**	**18**	**7**	**1**	**30**

***Industry Taught Course – II**

List of MOOCs and Vocational Courses will be published by the department before the commencement of respective semester.

Program: B.TECH. (Computer Engineering) Semester – V 2021 Course

Sr. No.	Name of Course	Teaching Scheme (Hrs./Week)			Examination Schemes (Marks)						Credits			
		L	P	T	UE	IA	TW	TW & OR	TW & PR	Total	L	P (TW/PR/OR)	T	Total
1	Algorithm Design and Analysis	3	2	-	60	40	-	-	50	150	3	1	-	4
2	Computer and Information Security*	3	2	-	60	40	-	50	-	150	3	1	-	4
3	Compiler Engineering	4	-	-	60	40	-	-	-	100	4	-	-	4
4	Data Warehousing and Mining	3	2	-	60	40	-	-	50	150	3	1	-	4
5	Microprocessors and Microcontrollers	3	2	-	60	40	-	-	50	150	3	1	-	4
6	Programming Technologies and Tools Laboratory – V	-	4	-	-	-	-	-	50	50	-	2	-	2
7	Project-I Stage- I	-	2	-	-	-	-	100	-	100	-	2	-	4
8	Vocational Course- III	-	-	-	-	-	-	-	-	-	-	-	-	2
9	MOOC-II	-	-	-	-	-	-	-	-	-	-	-	-	2
		16	**14**		**300**	**200**	**-**	**150**	**200**	**850**	**16**	**8**	**-**	**30**

*** Industry Taught Course – III**

List of MOOCs and Vocational Courses will be published by the department before the commencement of respective semester.

Program: B.TECH. (Computer Engineering) Semester – VI 2021 Course

Sr. No.	Name of Course	Teaching Scheme (Hrs./Week)			Examination Schemes (Marks)						Credits			
		L	P	T	UE	IA	TW	TW & OR	TW & PR	Total	L	P (TW/PR/OR)	T	Total
1	Big Data Analytics	4	2	-	60	40	-	-	50	150	4	1	-	5
2	Essentials of Internet of Things	3	2	-	60	40	-	50	-	150	3	1	-	4
3	Graphics Techniques and GPU	3	2	-	60	40	-	50	-	150	3	1	-	4
4	Mobile Architecture and Programming*	4	2	-	60	40	-	-	50	150	4	1	-	5
5	Quantitative Techniques, Communication and Values	2	2	-	60	40	-	-	-	100	3	-	-	3
6	Project-I Stage- II	-	2	-	-	-	-	100	-	100	-	4	-	4
7	Internship	-	-	-	-	-	-	50	-	50	-	3	-	3
8	Vocational Course- IV	-	-	-	-	-	-	-	-	-	-	-	-	2
		16	**12**		**300**	**200**	**-**	**250**	**100**	**850**	**17**	**11**	**-**	**30**

*** Industry Taught Course – IV**

List of MOOCs and Vocational Courses will be published by the department before the commencement of respective semester.

Program: B.TECH. (Computer Engineering) Semester – VII 2021 Course

Sr. No.	Name of Course	Teaching Scheme (Hrs./Week)			Examination Schemes (Marks)						Credits			
		L	P	T	UE	IA	TW	TW & OR	TW & PR	Total	L	P (TW/PR/OR)	T	Total
1	Artificial Intelligence*	4	2	-	60	40	-	-	100	200	4	1	-	5
2	Virtualisation and Cloud Computing	4	2	-	60	40	-	50	-	150	4	1	-	5
3	Scalable Computing	4	2	-	60	40	-	50	-	150	4	1	-	5
4	Elective - I	4	-	1	60	40	-	-	-	100	4	-	1	5
5	Programming Technologies and Tools Laboratory – VI	-	4	-	-	-	-	-	50	50	-	2	-	2
6	Project-II Stage - I	-	4	-	-	-	-	200	-	200	-	4	-	4
7	MOOC-III	-	-	-	-	-	-	-	-	-	-		-	2
8	Research Paper Publication	-	-	-	-	-	-	-	-	-	-		-	2
		16	**14**	**1**	**240**	**160**	**-**	**300**	**150**	**850**	**16**	**9**	**1**	**30**

*** Industry Taught Course – V**

Elective – I	Software Testing and Quality Assurance	Mobile Operating System	Fundamentals of Fog and Edge Computing	System Thinking

Program: B.TECH. (Computer Engineering) Semester – VIII 2021 Course

Sr. No.	Name of Course	Teaching Scheme (Hrs./Week)			Examination Schemes (Marks)						Credits			
		L	P	T	UE	IA	TW	TW & OR	TW & PR	Total	L	P (TW/PR/OR)	T	Total
1	Machine Learning*	4	2	-	60	40	-		100	200	4	1	-	5
2	Data Storage Networking	4	2	-	60	40	-	50		150	4	1	-	5
3	Data Visualisation	4	2	-	60	40	-	-	50	150	4	1	-	5
4	Elective – II	4	-	1	60	40	-	-	-	100	5	-	-	5
5	Programming Technologies and Tools Laboratory – VII	-	4	-	-	-	-	-	50	50	-	2	-	2
6	Project-II Stage - II	-	4	-	-	-	-	200	-	200	-	6	-	6
7	Social Activities - II	-	-	-	-	-	-	-	-	-	-	-	-	2
		16	**14**	**1**	**240**	**160**	**-**	**250**	**200**	**850**	**17**	**11**	**-**	**30**

*** Industry Taught Course – VI**

Elective – II	Intelligent Autonomous Systems & Robotics	Deep Learning	Blockchain and Cryptocurrencies	Docker and Kubernetes

B.TECH. (Computer Engineering)

SEMESTER – I

COURSE SYLLABUS

Mathematics for Computing - I

Teaching Scheme		Examination Scheme		Credit Scheme	
	Hours/Week		Marks		Credits
Lecture:	03 Hours/Week	University Examination	60 Marks	Lecture	03
Tutorial:	01 Hours/Week	Internal Assessment	40 Marks	Tutorial	01
		Total	100 Marks	Total	04

Course Objectives:

- Linear equations and its basis and dimension.
- Linear mapping and its matrix representation.
- Orthogonalization and diagonalization of matrices.

Prerequisite:
Knowledge of algebra of matrices and determinants

Course Outcomes: On completion of the course, students will have the ability to:

1. Apply rank of matrix in solving system of equations.
2. Identify basis and dimension of matrix.
3. Solve problems on kernel and image of linear transformation.
4. Apply linear operator to represent matrix.
5. Evaluate orthogonalization of inner product space.
6. Use methods to find eigen values and eigen vectors.

Unit I System of Linear Equation **06 Hours**

Vectors and linear combinations, Rank of a matrix, Gaussian elimination, LU Decomposition, Solving Systems of Linear Equations using the tools of Matrices.

Unit II Vector Spaces **06 Hours**

Definition, linear combination, spanning sets subspaces, linear dependence and independence, basis and dimension, rank of matrix.

Unit III Linear Mapping **06 Hours**

Linear mapping, Kernel and image of linear mapping, rank and nullity of a linear mapping, singular and non-singular linear mapping.

Unit IV Linear mapping and matrices **06 Hours**

Matrix representation of linear operator, change of base, similarity matrices

Unit V Inner Product space and orthogonalization **06 Hours**

Inner product space, Cauchy-schwarz equality, Orthogonality, Orthogonal sets and bases, projections, Gramschidt orthogonalization, orthogonal and positive definite matrices, matrix representation of inner product

Unit VI Diagonalisation: Eigen values and eigen vectors **06 Hours**

Characteristic polynomial, Cayley-Hamilton theorem, eigen values and eigen vectors, properties.

Textbooks

1. P. N. Wartikar and J. N. Wartikar, Applied Mathematics (Volumes I and II), 7th Ed., Pune Vidyarthi GrihaPrakashan, Pune, 2013.
2. B. S. Grewal, Higher Engineering Mathematics, 42nd Ed., Khanna Publication, Delhi

3. B.V. Ramana, Higher Engineering Mathematics, 6th Ed., Tata McGraw-Hill, New Delhi, 2008.
4. Erwin Kreyszig, Advanced Engineering Mathematics, 10th Ed., John Wiley & Sons, Inc., 2015.

Reference Books

1. Peter V. O'Neil, Advanced Engineering Mathematics, 7th Ed., Cengage Learning, 2012.
2. Michael Greenberg, Advanced Engineering Mathematics, 2nd Ed., Pearson Education, 1998.

Project Based Learning - Provisional List of Projects

Students are expected prepare report on any one topic, write its definition, applications and illustrate with few examples. Also, write pseudo code for it, wherever applicable.

1. Gauss Elimination method.
2. LU-decomposition method
3. Rank of matrix
4. Linear combination
5. Basis and dimension
6. Spanning sets
7. Kernel and image of linear transformation
8. Rank-nullity theorem
9. Non-singular linear mapping
10. Linear operator
11. Similarity matrices
12. Change of base
13. Cauchy Schwarz equality
14. Orthogonality
15. Gram Schmidt Orthogonalization
16. Matrix representation of matrix
17. Cayley-Hamilton theorem
18. Eigen values and Eigen vectors

(Note: - *Students in a group of 3 to 4 shall complete any one project from the above list)

Syllabus for Unit Tests:

Unit Test -1	Unit – I, Unit – II, Unit - III
Unit Test -2	Unit – IV, Unit – V, Unit - VI

Organic and Electrochemistry

Teaching Scheme		Examination Scheme		Credit Scheme	
	Hours/Week		Marks		Credits
Lecture:	03 Hours/Week	University Examination	60 Marks	Lecture	03
Practical:	02 Hours/Week	Internal Assessment	40 Marks		
		Term work	25 Marks	Practical	01
		Total	125 Marks	Total	04

Course Objectives:
The student should acquire the knowledge of

- To develop the interest among the students regarding chemistry and their applications in engineering.
- To develop confidence among students about chemistry, how the knowledge of chemistry is applied in technological field.
- The student should understand the concepts of chemistry to lay the groundwork for subsequent studies in the computing field.

Prerequisite:
Basic Chemistry

Course Outcomes: On completion of the course, students will have the ability to:

1. Differentiate between ionic and covalent bonding and classify the bonding in a compound as ionic or covalent.
2. Develop a working knowledge of the twelve fundamental principles of green chemistry and what it is all about.
3. Apply standard reduction potential data to determine the relative strength of oxidizing/reducing agents
4. Demonstrate the knowledge of polymer materials for futuristic engineering applications
5. Describe the properties of materials and Application of semiconductor electronics
6. Describe the manufacturing and refining process of fuels and lubricants

Unit I Chemical Bonding in Molecules **06 Hours**

MO theory, Structure, bonding and energy levels of bonding and shapes of many atom molecules, Coordination Chemistry, Electronic spectra and magnetic properties of complexes with relevance to bio-inorganic chemistry, organometallic chemistry.

Unit II Green Chemistry **06 Hours**

Introduction, Twelve Principles of Green chemistry, numerical on atom economy, synthesis, adipic acid and indigo. Organic dye- Traditional methods of organic dye. Green solvents (ionic liquid supercritical CO2), and products from natural materials.

Unit III Electrochemistry **06 Hours**

Electrochemical cells and Galvanic cells, EMF of a cell, Single electrode potential, Nernst equation, Electrochemical series, Types of electrodes, Reference electrodes, pH, pOH, acids and basis, Fuel cells, Construction and Working of - Acid and Alkaline Storage Battery, Dry Cell, Ni-Cd Batteries, Li-Ion Batteries, Li-Po Batteries.

Unit IV Polymers for The Electronics Industry **06 Hours**

Polymers, Conduction mechanism, Preparation of conductive polymers, Polyacetylene, Poly (p- phenlylene), Polyhetrocyclic systems, Polyaniline Poly (Phenylene sulphide), Poly (1,6-heptadiyne), Applications, Photonic applications.

Unit V Semi-Conductors, Insulators and Superconductors **06 Hours**

Semi conductivity in non-elemental materials, Preparations of semiconductors, Chalcogen photoconductors, photocopying process Introduction to Superconductors, types of Superconductors, Properties of superconductors, Applications of Superconductors, Electrical insulators, or Dielectrics.

Unit VI Fuels & Lubricants **06 Hours**

Classification of fuels, Calorific values, Comparison between solid, liquid, and gaseous fuels, Theoretical calculation of calorific value of a fuel, Selection of coal, analysis of coal, Natural Gas, Producer gas, water gas, Lubricants, Mechanism of lubrication, classification of lubricants, lubricating oils, Solid lubricants, Greases or Semi-Solid lubricants, Synthetic lubricants, Lubricating emulsions, Properties of lubricating oils.

Textbooks

1. Polymer Science and technology (2nd Edition), P. Ghosh, Tata McGRAW Hill, 2008.
2. Polymers: Chemistry & Physics of Modern Materials (2nd edition) J.M.G.Cowie, Blackie Academic & Professional, 1994.
3. A Textbook of Engineering Chemistry, Shashi Chawla, Dhanpat Rai & Co, 2004
4. Engineering Chemistry (16th Edition) Jain, Jain, Dhanpat Rai Publishing Company, 2013.

Reference Books

1. Inorganic Chemistry (4th edition), D. F. Shrives and P. W. Atkins, Oxford University, Oxford, 2006.
2. Reactions, Rearrangements and Reagents (4th edition), S. N. Sanyal, Bharti Bhawan (P & D), 2003.
3. Applications of Absorption Spectroscopy of Organic Compounds (4th edition), John R. Dyer, Prentice Hall of India Pvt. Ltd., 1978.

List of Assignments

Six assignments to be given by the course coordinator (Theory)-one from each unit/one mini project with report-students can work in group of 4 Maximum

List of Laboratory Exercises

1. Determination of Hardness of water sample by EDTA method.
2. Determination of Chloride content in water sample by precipitation titration method.
3. To determine strength of acid by pH – metric Titration
4. To measure the Conductance of a solution by conductometric titration
5. Measurement of Surface tension of a given liquid by Stalagmometer.
6. Determination of viscosity of a given liquid by Ostwald's Viscometer.
7. Determination of Saponification value of an oil sample.
8. To determine alkalinity water sample
9. Determination of Hardness of water sample by EDTA method.
10. Determination of Chloride content in water sample by precipitation titration method

11. To determine strength of acid by pH – metric Titration
12. To Prepare Phenol formaldehyde/Urea formaldehyde resin
13. To study set up of Daniel cell.

Project Based Learning - Provisional List of Projects

1. Green Chemistry approach to Nano-Structured Electronics
2. Assessment of Environmentally Benign Photopolymers as an Alternative to the Use of Formaldehyde Based Textile Finishing Agents
3. Solvent-Free Synthesis of Phthalocyanines
4. Synthesis of Conjugated Polymers and Molecules Using Sugar Reagents and Solventless Reactions
5. Environmentally Benign Control of Polymer Solubility: Photoresist Materials Using DNA Mimics
6. Enzymatic Synthesis of Non-Formaldehyde Phenolic Polymers: Control of Hydrogen Peroxide Concentration.
7. The materials chemistry and electrochemistry of lithium and sodium-ion batteries
8. Electroplating- the principles, how different metals can be used and the practical applications
9. Electroplating, Metal Polishing, Anodizing, Phosphating Metal Finishing and Powder Coating Projects.
10. To determine calorific value of a fuel by any suitable method
11. To study various properties of lubricants
12. To study various types of lubricants and its properties.
13. To determine quality of coal sample & its analysis.
14. To study mechanism of lubrication.
15. To study coal analysis & its significance.

Note: - Students in a group of 3 to 4 shall complete any one project from the above list)

Syllabus for Unit Tests:

Unit Test -1	Unit – I, Unit – II, Unit - III
Unit Test -2	Unit – IV, Unit – V, Unit - VI

Digital Electronics

Teaching Scheme		Examination Scheme		Credit Scheme	
	Hours/Week		Marks		Credits
Lecture:	04 Hours/Week	University Examination	60 Marks	Lecture	04
Practical:	02 Hours/Week	Internal Assessment	40 Marks		
		Term work	25 Marks	Practical	01
		Total	125 Marks	Total	05

Course Objectives:

- To present the Digital fundamentals, Boolean algebra, and its applications in digital systems
- To familiarize with the design of various combinational digital circuits using logic gates
- To introduce the analysis and design procedures for synchronous and asynchronous sequential circuits
- To understand the various semiconductor memories and related technology

Prerequisite:
Mathematics and Elementary Physics

Course Outcomes: On completion of the course, students will have the ability to:

1. Comprehend different number systems and Boolean algebraic principles.
2. Apply logic design minimization techniques to simplify Boolean expressions
3. Analyse and design combinational logic circuits.
4. Demonstrate the operations of systems with sequential circuit elements.
5. Comprehend characteristics and structure of Programmable Logic Devices and Memory.
6. Draw ASM charts for sequential circuit design.

Unit I Digital systems **08 Hours**

Number Systems: Introduction to Number Systems-Decimal, Binary, Octal, Hexadecimal, Conversion of number system, Representation of Negative Numbers, 1's complement and 2's complement.
Binary Arithmetic: Binary addition, Binary subtraction, Subtraction using 1's complement and 2's complement, Binary multiplication, and division.
Digital Codes: BCD code, Excess-3 code, Gray code and ASCII code.
Logic Gates: Logical Operators, Logic Gates-Basic Gates, Universal Gates, realization of other gates using universal gates.

Unit II Logic Design Minimization **08 Hours**

Boolean algebra, De Morgan's Theorems, Standard representation of logic functions, Sum of Product (SOP) form, Product of Sum (POS) form, Simplification of logical functions, Minimization of SOP and POS forms using Karnaugh-Maps up to 4 variables Don't care condition, Quine-McCluskey Method.

Unit III Combinational Circuits **08 Hours**

Binary and BCD arithmetic, Half Adder, Full Adder, Half Subtractor, Full Subtractor, Binary Adder (IC 7483), BCD adder, Code converters
Multiplexers, De multiplexer, Decoder (IC 74138) and their use in combinational logic design, Priority Encoder, Digital Comparators, Parity generators and Checker (IC 74180), ALU.

Unit IV Sequential Circuits **08 Hours**

Flip- flop: SR, JK, D, T flip flops, Truth Tables and Excitation tables, Conversion from one type to another type of Flip Flop.
Registers: Buffer register, Shift register.
Counters: Asynchronous counters, Synchronous counters, Modulus counters

Unit V FSM and ASM charts **08 Hours**

Introduction to FSM, Moore and Mealy State machine, state machine as a sequential controller. Design of state machines: state table, state assignment, transition/excitation table, excitation maps and equations, logic realization, ASM chart notations, ASM block, State diagram, ASM chart for sequential circuits, Multiplexer Controller.

Unit VI Memory and PLD: **08 Hours**

Semiconductor memories: memory organization, memory expansion, Classification and characteristics of memories, RAM, ROM, EPROM, EEPROM, NVRAM, SRAM, DRAM.
Programmable logic devices: Study of PROM, PAL, PLAs. Architecture of PLA, designing combinational circuits using PLDs.

Textbooks

1. M. Morris Mano and M. D. Ciletti, Digital Design, Pearson Education.
2. RP Jain, Modern Digital Electronics, Tata McGraw Hill Publication.
3. F.J. Hill and G.L. Peterson, Switching Theory and Logic Design, John Wiley
4. J.F.Wakerly "Digital Design: Principles and Practices", 3rd edition, 4th reprint, Pearson Education, 2

Reference Books

1. David J. Comer, Digital Logic & State Machine Design, Oxford University Press.
2. Digital Integrated Electronics- H.Taub & D.Shilling, Mc Graw Hill.

List of Assignments

Six assignments to be given by the course coordinator (Theory)-one from each unit

List of Laboratory Exercises

1. Verify truth tables of logic gates. (AND, OR, XOR, NOT, NAND, NOR). Simplify the given Boolean expression using K-map and implement using gates
2. State De-Morgan's theorem and write Boolean laws. Implement NAND and NOR as Universal gates.
3. Design (truth table, K-map) and implement half and full adder/ subtractor.
4. Design (truth table, K-map) and implement 4-bit BCD to Excess-3 Code converters.
5. Study of magnitude Comparator using IC 7485
6. Implement of logic functions using multiplexer IC 74151 (Verification, cascading & logic function implementation)
7. Implement logic functions using 3:8 decoder IC 74138.
8. Verify truth tables of different types of flip flops.
9. Design (State diagram, state table & K map) and implement 3 bits Up and Down Asynchronous and Synchronous Counter using JK flip-flop
10. Design and implement modulo 'n' counter with IC 7490.

Project Based Learning - Provisional List of Projects

1. Survey report of basic gates ICs 7432, 4011, 4050, 4070,4071,40106
2. Implement combinational logic Circuit of given Boolean Equation.
3. Implement Half Adder and Half Subtractor.

4. Implement Full Adder using two Half Adders
5. Build 4-bit parallel Adder / Subtractor using IC.
6. Build Code Converters: Binary to Gray
7. Build Code Converters: Excess 3 to Binary)
8. Implement Two Bit Magnitude Comparator using IC 7485
9. Implement given combinational logic using MUX
10. Implement 7 segment decoder driver using IC 7447.
11. Build a Decade counter and Up-Down Counter.
12. Build a Shift Registers: SISO and SIPO
13. Implement the Johnson Counter and Ring Counter.
14.Survey Report on Static I/O and transfer Characteristic of TTL and CMOS.
15. Implement given Boolean Function using PLA. (Function and Equation will be given by Subject Teacher)

Syllabus for Unit Tests:

Unit Test -1	Unit – I, Unit – II, Unit - III
Unit Test -2	Unit – IV, Unit – V, Unit - VI

Classic Data Structures

Teaching Scheme		Examination Scheme		Credit Scheme	
	Hours/Week		Marks		Credits
Lecture:	04 Hours/Week	University Examination	60 Marks	Lecture	04
Practical:	02 Hours/Week	Internal Assessment	40 Marks		
		Term Work & Practical	50 Marks	Practical	01
		Total	150 Marks	Total	05

Course Objectives:

The course focuses on enabling students to understand how data is stored in computer programs using data structures and facilitate them to use and build fundamental data structures.

Prerequisite:

Programming Basics

Course Outcomes: On completion of the course, students will have the ability to:

1. Compare and contrast the interfaces and internal representation of several linear abstract data types.
2. Solve given problems using array
3. Implement Stacks in a high-level programming language
4. Use and Implement Queues in a high-level programming language
5. Use and Implement lists in a high-level programming language
6. Demonstrate the ability to analyse, design, apply and use data structures and algorithms to solve engineering problems.

Unit I Introduction to Data structures & Arrays **08 Hours**

Need of Data structure, Classification of Data Structures, Static Data Organization, Operations on Data Structures, Abstract data Types (ADT).
Arrays: Introduction, Array Operations, representation of Arrays in Memory, Array with Functions, One- & Two-dimensional array in function, Implementation of One- & Two-Dimensional Arrays in Memory.
Applications: string handling, polynomial equation solving, sparse matrix multiplication, tic-tac-toe, and data visualization

Unit II Stacks **08 Hours**

Stack Definition and Structure, Operations on Stacks – create stack, Push stack, Pop stack, Stack top, Empty Stack, stack count, Destroy Stack, Array and Linked Representation, Types of Notations – Prefix, Infix and Postfix,
Applications of Stack: Reversing Data, Converts Decimal to Binary, Parsing, Postponement, expression Conversion, and evaluation.

Unit III Queue **08 Hours**

Queue: Introduction, Definition, ADT for queue, Storage Methods, Queue Operations, Enqueue, Dequeue, Queue front, Queue rear, Queue Example, Create Queue, priority Queue, Circular Queue.
Application of Queue: Categorising Data, Queue Simulation.

Unit IV Linear Lists **08 Hours**

Introduction, singly linked list, Circularly Linked List, Doubly Linked lists, Basic operations, - Insertion, Deletion, retrieval, traversal, create List, insert node, delete node, List Search, Empty list, Destroy list.

Unit V Linked Stacks and Linked Queues **08 Hours**

Introduction, Operations on Linked stacks and Linked Queues, Dynamic Memory management and Linked Stacks, Implementation of Linked Representations.

Unit VI Overview of Real time Applications of Linear Data Structures **08 Hours**

Stacks – Balancing of Symbols, Infix to Postfix, Evaluation of Postfix expression, Implementing Function Calls, Finding of Spans, undo sequence in text editor, Matching Tags in HTML and XML.
Linked List – Implement Stack using Linked List.
Queues – Scheduling Jobs, Simulation of real-world queues such as ticket counter or first come first served scenarios, Asynchronous Data Transfer.

Textbooks

1. Brassard & Bratley, ―Fundamentals of Algorithmics‖, Prentice Hall India/Pearson Education, ISBN 13-9788120311312.
2. Horowitz and Sahani, ―Fundamentals of Data Structures in C++‖, University Press, ISBN 10: 0716782928 ISBN 13: 9780716782926.
3. Goodrich, Tamassia, Goldwasser, ―Data Structures and Algorithms in C++‖, Wiley publication, ISBN-978-81-265-1260-7
4. Data Structure and Algorithmic Thinking with Python, CareerMonk Publications, Narasimha Karumanchi, 2016

Reference Books

1. Richard F Gilberg & Behrouz A Forouzan, Data Structures (A Pseudocode Approach with C), second edition, Cengage Learning, 2004.
2. PAI, Data Structures, Tata McGraw-Hill Education, 2008
3. Mayank Patel, Data Structure and Algorithm With C, Edu creation Publishing, 2018
4. Thomas H. Cormen, Charles E Leiserson, Ronald L Rivest, Clifford Stein, Introduction to Algorithms, MIT Press, 2001.

List of Assignments

1. Show how you can efficiently implement one stack using two queues.
2. What is the most appropriate data structure lo print elements of queue in reverse order?
3. You are given a pointer to the first element of a linked list L. There are two possibilities for L, it either ends (snake) or its last clement points back to one of the earlier elements in the list (snail). Give an algorithm that tests whether a given list L is a snake or a snail.
4. Check whether the given linked list is either NULL-terminated or ends in a cycle (cyclic).
5. Find nth node from the end of a Linked List
6. Simulate real world queues such as ticket counter.

List of Laboratory Exercises

1. Study assignment on programming IDE Tools
2. Write a program to implement one dimensional array.
3. Write a program to design tic-tac-toe game
4. Write a program to perform basic operation on stack.
5. Write a program to convert and evaluate polish notations.
6. Write a program to perform basic operation on stack.
7. Write a program to implement Priority queue & Double Ended Queue.
8. Write a program to perform basic operation on circular queue.
9. Write a program to implement hashing technique.

10. Write a program to implement searching and sorting techniques

Project Based Learning - Provisional List of Projects

1. Expression Evaluation
2. Traffic Management System
3. Library Management System
4. Employee Record System
5. Dictionary
6. Calendar Application
7. Medical Store Management System
8. Cricket Score Sheet
9. Bank Management System
10. Telephone directory

(Note:- *Students in a group of 3 to 4 shall complete any one project from the above list)

Syllabus for Unit Tests:

Unit Test -1	Unit – I, Unit – II, Unit - III
Unit Test -2	Unit – IV, Unit – V, Unit - VI

Computational Thinking and Programming Concepts

Teaching Scheme		Examination Scheme		Credit Scheme	
	Hours/Week		Marks		Credits
Lecture	04 Hours/Week	University Examination	60 Marks	Lecture	04
Practical	02 Hours/Week	Internal Assessment	40 Marks		
		Term Work & Practical	100 Marks	Practical	01
		Total	200 Marks	Total	05

Course Objective

The aim of this course is to make students to think in a computational manner to a point where they can derive simple algorithms and code the programs to solve some basic problems in their domain of studies.

Prerequisite:

-

Course Outcomes: On completion of the course, students will have the ability to:

1. Formulate a problem and express its solution in such a way that a computer can effectively carry it out.
2. Apply the Computational Thinking (CT) concepts on case studies/problem-based scenarios through hands-on practice of the CT processes.
3. Write algorithm and pseudo code for the identified strategy
4. Use Abstraction and Modelling.
5. Solve given problems through scratch based graphical programming tool
6. Demonstrate logical and algorithmic thinking.

Unit I Overview of Problem Solving **08 Hours**

Problem Solving Concepts: Formal Problem Definition, Challenges in Problem Solving, Problem solving with Computers, Framework for Problem Solving.
Introduction to Problem solving tools: Flowcharts, algorithm, pseudocode, Data structures.

Unit II Logical and Algorithmic Thinking **08 Hours**

Inductive Vs Deductive arguments, Logic, Boolean Logic, Symbolic Logic, Logical operators and their symbols, Propositional Logic
Algorithmic Thinking: Algorithms, Intuition vs precision, defining algorithms, Algorithm constructs, Controlling algorithm execution, Complex conditionals.

Unit III Overview of Computational Thinking **08 Hours**

About Computational Thinking, Data Representation and Abstraction - Problem formulation, Devising a Solution, Decomposition, Pattern recognition, Generalisation, Evaluation.

Unit IV Overview of Programming Concepts **08 Hours**

Scratch Programming – Working of Scratch, Scratch tool, Motions and Drawing, Looks and Sound, Procedures, Variables, Making decisions, Loops, String Processing, Lists.
Introduction to higher level programming languages like C, Python, C++ and its constructs.

Unit V Limits of Computation **08 Hours**

Capacity Measurement in Computers, Estimate of Physical limitations, Benchmarks, Counting the performance, impractical algorithms, Metaphysical limitations, Impossible algorithms.

Unit VI Computational Thinking in Software Development **08 Hours**

Effective Building Blocks: Basic Algorithms Constructs, Program State, Code Organization, Using Abstractions and Patterns, Effective Modelling: Objectives, Entities, Relationship, Processes, Usage and General Advice.
Testing and Evaluating Programs, Anticipating Bugs, Syntax vs semantic errors, Defensive programming, Verification and validation, Testing the Parts, Testing the Whole, Debugging Case Study: Home Automation System

Textbooks

1. Computational Thinking, By Peter J. Denning and Matti Tedre, The MIT Press Essential Knowledge series
2. Computational Thinking and Coding for Every Student, Jane Krauss, Kiki Prottsman by Corwin Publishers
3. Computational Thinking for the modern problem solver, David D riley, Kenny A Hunt, CRC Press, 2014
4. Computational thinking a beginner's guide to problem solving and programming, Karl Beecher, BCS Learning & Development, 2017

Reference Books

1. How to Solve it by Computer by R. G. Dromey, 1e, Pearson Education.
2. Learn to program with Scratch, Majed Marji, no starch press, 2014
3. Let Us C, Yashavant Kanetkar, Infinity Science Press, 2008
4. Let Us C++, Yashavant Kanetkar, BPB Publications, 1999
5. Introduction to Computation and Programming Using Python, Mit Press, John Guttag, 2016

List of Assignments

1. The Following problems can be solved using SCRATCH Tool: Create a function block that calculates the force needed to accelerate 2,000 kg car 3 m/s2
2. Write different procedures to draw each letter of your name. Name each procedure for the letter that it draws. Then write a script that calls these procedures so you can draw your name on the Stage
3. Write a program that prompts the user to enter five test scores between 1 and 10. The program will then count the number of scores that are greater than 7
4. The Pythagorean theorem states that if a and b are the lengths of the legs of a right triangle and c is the length of the hypotenuse (the longest side), then a2 + b2 = c2 . Write a program that gets three numbers from the user and determines whether they could represent the sides of a right triangle.
5. Create two lists for storing the items sold in a grocery store and their corresponding prices. Write a program that asks the user to enter an item's name and then displays that item's price, if it is found in the list.
6. Write a program that prompts the user to enter the highest and lowest temperatures for the 12 months of a year. Store the input values in two lists.

List of Laboratory Exercises

1. WAP to SWAP (interchange) 2 numbers without using third variable
2. WAP to find the sum and average of values appearing at the positions divisible by 3 in the given sequence of n values
3. WAP that receives any year from the keyboard and uses a function to determine whether the year is a leap year or not.
4. WAP that uses a function that converts a lowercase character to its uppercase
5. WAP to read n numbers and count even and odd numbers.

6. WAP that uses a recursive function to convert given decimal number into its binary equivalent.
7. WAP to use the suitable function to obtain the prime factors recursively.
8. WAP that uses a function that prints the nth element of Fibonacci series using recursion method.
9. WAP that uses a function to calculate the sum of n odd integers.
10. WAP that uses a function power that calculates the power of a given number.

Project Based Learning - Provisional List of Projects

1. Identify any patterns in the problem.
2. Build Model for various Mathematical Formulas
3. Study the friendship link of any social networking site.
4. Using primary data source study, the voting pattens of our country.
5. Analyse how algorithms effect social media feeds
6. Visualize and Interpret performance of Athlete for any Sport
7. Modularize a given problem into sub problems.
8. Analyse the next moves of a player for Game of Chess
9. Devise a strategy to compute Result of a particular Class
10. Library Management System

Syllabus for Unit Tests:

Unit Test -1	Unit – I, Unit – II, Unit - III
Unit Test -2	Unit – IV, Unit – V, Unit - VI

Programming Technologies and Tools Laboratory – I

Teaching Scheme		Examination Scheme		Credit Scheme	
	Hours/Week		Marks		Credits
Practical:	04 Hours/Week	Term Work & Practical	50 Marks	Practical	02
		Total	50 Marks	Total	02

Course Objective

The course is designed to provide complete knowledge of C language. Students will be able to develop logics which will help them to create programs, applications in C. Also, by learning the basic programming constructs they can easily switch over to any other language in future.

Prerequisite:

-

Course Outcomes: On completion of the course, students will have the ability to:

1. Demonstrate the knowledge of C programming Concepts
2. Develop C programs
3. Define Data types and use them in data processing programs.
4. Trace the execution of programs written in C language
5. Write functions and implement.
6. Analyse and interpret the concept of declarations, initialization, operations on pointers and their usage.

Unit I Basics and Operators **08 Hours**

General problem-solving strategies, Top-down design, Introduction to program Planning tools- algorithm, flowcharts, and pseudo codes. Introduction to Logic Structures: Sequential structure, Decision Structure, Loop Structure. Features of C, basic concepts, structure of C program, program, declarations, variables, data types, expressions, operator's assignment, arithmetic, relational, logical, increment and decrement, precedence of operators, type conversions, scanf and printf functions

Unit II Control structures **08 Hours**

if-else, nested if-else, cascaded if-else and switch statement. C Conditional control structures: for, while do-while Unconditional control structures: break, continue, goto statement.

Unit III Arrays and strings **08 Hours**

Declaration initialization of one-dimensional Array, two-dimensional array, accessing array elements, Character Array/String, Character - Handling Library Functions, Standard Input/Output Library Functions for string.

Unit IV Functions and structures **08 Hours**

What is a Function, Benefits of a Function, Function Terminology, Array of Structures, How does Function Works , Scope and Lifetime of Variables in function ,Storage Classes of Variables , Call by value and call by reference ,Recursion ,Overview of Structures , Defining and Using a Structure , Structures within a Structure

Unit V Pointers **08 Hours**

Declaring and Initializing Pointers, Function and Pointer Parameters, Pointer Arithmetic, Pointer and Arrays, Two Dimensional Arrays and Pointers.

Unit VI Files **08 Hours**

FILE, Opening and Closing of Files, Writing and Reading in Text Format, Writing and Reading in Binary Format, Command Line Arguments

T**extbooks**

1. Let Us C by Yashavant Kanetkar, 13e, BPB Publication.
2. Brain W.Kernighan & Dennis Ritchie, C Programming Language, 2nd edition, PHI
3. E.Balaguruswamy, Programming in ANSI C 5th Edition McGraw-Hill
4. How to Solve it by Computer by R. G. Dromey, 1e, Pearson Education.

Reference Books

1. C: The Complete Reference by Herbert Schildt.

List of Laboratory Exercises

1. Write a program to read a four-digit integer and print the sum of its digits.
2. Use recursive function calls to evaluate F(x) = x – x3 / 3! + s5/5! –x7/7! + ……
3. WAP to print the table of n.
4. Write a 'C' Program to evaluate Ackerman Function
5. Given a list of marks ranging from 0 to 100, write a program to compute and print the number of students:
   (a) who have obtained more than 80 marks,
   (b) who have obtained more than 60 marks,
   (c) who have obtained more than 40 marks,
   (d) who have obtained 40 or less marks,
   (e) in the range 81 to 100,
   (f) in the range 61 to 80,
   (g) in the range 41 to 60, and
   (h) in the range 0 to 40.
6. Make a Book Shop Inventory. The list should include details such as author, title, price, publisher, stock position. When a particular title and author name is given as input the program should reply whether it is in the list or no. If not, appropriate message should be displayed.
7. Write a program to find the total number of characters in a file.
8. Write a function which takes to integer as argument and return their average in float. WAP to test this function.
9. WAP to read n numbers and count even and odd numbers
10. Write a function which takes to integer as argument and return their sum. WAP to test this function.

Project Based Learning - Provisional List of Projects

1. Hangman Game
2. Modern Periodic Table
3. Pacman Game
4. Personal Diary Management System
5. Phonebook Application
6. Quiz Game
7. School Billing System
8. Snake Game
9. Telecom Billing System
10. Tic-Tac-Toe Game
11. Typing Tutor

B.TECH (Computer Engineering)

SEMESTER – II

COURSE SYLLABUS

Mathematics for Computing - II

Teaching Scheme		Examination Scheme		Credit Scheme	
	Hours/Week		Marks		Credits
Lecture:	03 Hours/Week	University Examination	60 Marks	Lecture	03
Tutorial:	01 Hours/Week	Internal Assessment	40 Marks	Tutorial	01
		Total	100 Marks	Total	04

Course Objectives:

To equip students with knowledge of:

- Fourier series and integral transforms.
- Multiple integrals and its applications.
- Vector calculus and its applications.

Prerequisite:

The students should have knowledge of vector algebra, derivative, and integration.

Course Outcomes: On completion of the course, students will have the ability to:

1. Use periodic functions as fourier series.
2. Apply methods of finding fourier and Z-transforms.
3. Apply methods of laplace transform of piecewise continuous functions.
4. Identify concepts of double and triple integrals.
5. Apply vector derivative for physical quantities.
6. Evaluate line, surface, and volume integrals.

Unit I Fourier Series **06 Hours**

Definition, Dirichlet's conditions, Fourier Series and Half Range Fourier Series, Harmonic Analysis

Unit II Fourier and Z-Transform **06 Hours**

Fourier Transform (FT): Complex Exponential Form of Fourier series, Fourier Integral Theorem, Sine & Cosine Integrals, Fourier Transform, Fourier Sine and Cosine Transform and their Inverses. Introductory
Z-Transform (ZT): Definition, Standard Properties, ZT of Standard Sequences and their Inverses. Solution of Simple Difference Equations.

Unit III Laplace Transform and its application **06 Hours**

Definition of LT, Inverse LT. Properties & theorems. LT of standard functions. LT of some special functions viz., Periodic, Unit Step, Unit Impulse, ramp, jump, Problems on finding LT & inverse LT. Applications of LT and Inverse LT for solving ordinary differential equations.

Unit IV Multiple Integrals and their Application **06 Hours**

Double and Triple integrations, Applications to Area, Volume, Mean and Root Mean Square Values, moment of inertia, centre of gravity

Unit V Vector Differential Calculus **06 Hours**

Physical interpretation of Vector differentiation, Vector differential operator, Gradient, Divergence and Curl, Directional derivative, Solenoidal, Irrotational and Conservative fields, Scalar potential, Vector identities.

Unit VI Vector Integral Calculus and Applications **06 Hours**

Line, Surface, and Volume integrals, Work-done, Green's Lemma, Gauss's Divergence theorem, Stoke's theorem. Applications to problem in engineering.

Textbooks

1. P. N. Wartikar and J. N. Wartikar, Applied Mathematics (Volumes I and II), 7th Ed., Pune Vidyarthi GrihaPrakashan, Pune, 2013.
2. B. S. Grewal, Higher Engineering Mathematics, 42nd Ed., Khanna Publication, Delhi
3. B.V. Ramana, Higher Engineering Mathematics, 6th Ed., Tata McGraw-Hill, New Delhi, 2008.

Reference Books

1. Erwin Kreyszig, Advanced Engineering Mathematics, 10th Ed., John Wiley & Sons, Inc., 2015.
2. Peter V. O'Neil, Advanced Engineering Mathematics, 7th Ed., Cengage Learning, 2012.

List of Assignments

Six assignments to be given by the course coordinator (Theory)-one from each unit

Project Based Learning - Provisional List of Projects

Students are expected prepare report on any one topic, write its definition, applications and illustrate with few examples. Also, write pseudo code for it, wherever applicable.

1. Fourier series
2. Harmonic analysis
3. Fourier transform
4. Z-Transform
5. Laplace transform technique to solve ODE
6. Multiple Integral to evaluate area and volume
7. Directional derivative
8. Divergence and curl
9. Greens theorem
10. Gauss Divergence Theorem
11. Stokes theorem
12. Unit step function
13. Solenoidal and irrotational fields
14. Simple difference equation
15. Periodic functions

Note: - *Students in a group of 3 to 4 shall complete any one project from the above list)

Syllabus for Unit Tests:

Unit Test -1	Unit – I, Unit – II, Unit - III
Unit Test -2	Unit – IV, Unit – V, Unit - VI

Physics for Computing Systems

Teaching Scheme		Examination Scheme		Credit Scheme	
	Hours/Week		Marks		Credits
Lecture:	03 Hours/Week	University Examination	60 Marks	Lecture	03
Practical:	02 Hours/Week	Internal Assessment	40 Marks		
		Term work	25 Marks	Practical	01
		Total	125 Marks	Total	04

Course Objective

To impart knowledge of basic concepts in physics relevant to engineering applications in a broader sense with a view to lay foundation for the Computer Engineering and Science.

Prerequisite:

Basic understanding of physics and calculus.

Course Outcomes: On completion of the course, students will have the ability to:

1. Interpret the properties of charged particles to develop modern instruments such as electron microscopy.
2. Appraise the wave nature of light and apply it to measure stress, pressure, and dimension etc.
3. Summarise the structure and properties of lasers to their performance and intended applications.
4. Classify the optical fibre, understanding the structure, types, and its applications in the field of communication.
5. Solve quantum physics problems to micro level phenomena and solid-state physics
6. Explain mechanical properties of solid matter and connect to applications in the field of engineering.

Unit I Modern Physics **06 Hours**

Motion of a charged particle in electric and magnetic fields, Electrostatic and Magnetostatic focusing, Electron microscope, Wavelength and resolution, Specimen limitation, Depth of field and focus, Transmission electron microscope (TEM), Scanning electron microscope (SEM), Separation of isotopes by Bainbridge mass spectrograph, Cathode ray tube (CRT).

Unit II Wave Optics **06 Hours**

Interference: Interference of waves, interference due to thin film (Uniform and nonuniform (only formula-no derivation is expected), Newton's ring, Applications of interference (optical flatness, highly reflecting films, non-reflecting coatings).

Diffraction: Introduction, Classes of diffraction, Diffraction at a single slit (Geometrical method), Conditions for maximum and minimum, Plane diffraction grating, Conditions for principal maxima and minima

Polarisation: Introduction, Double refraction and Huygen's theory, Positive and negative crystals, Nicol prism, Dichroism.

Unit III Lasers **06 Hours**

Principle of laser, Einstein's coefficients, Spontaneous and stimulated emission, Population inversion, Ruby laser, Helium-Neon laser, Semiconductor laser, Single Hetro-junction laser, Gas laser: CO2 laser, Properties of lasers, Laser speckles, Applications of lasers (Engineering/ industry, medicine, Computers)

Unit IV Fibre Optic **06 Hours**

Principle of fibre optics, Construction, Numerical Aperture for step index fibre; critical angle, angle of acceptance, V number, number of modes of propagation, types of optical fibres, Fibre optic communication system, advantages, and disadvantages of fibre optics.

Unit V Quantum Mechanics **06 Hours**

Dual nature of matter, DeBroglie's hypothesis, Heisenberg's uncertainty principle with illustrations, Physical significance of wave function, Schrodinger's time dependant and time independent wave equation, Application of Schrodinger's time independent wave equation to the problems of Particle in a rigid box, step potential and potential barrier (analytical discussion), tunnelling effect.

Unit VI Solid state physics **06 Hours**

Free electron theory, Density of states, Bloch theorem (Statement only), Origin of band gap, Energy bands in solids, Effective mass of electron, Fermi-Dirac probability function and position of Fermi level in intrinsic semi-conductors (with derivation) and in extrinsic semi-conductors, Band structure of p-n junction diode under forward and reverse biasing, Conductivity in conductor and semi-conductor, Hall effect and Hall coefficient, Photovoltaic effect, Solar cell, and its characteristics.

Textbooks

1. A Textbook of Engineering Physics, M N Avadhanulu, P G Kshirsagar and TVS Arun Murthy, S. Chand Publishing (2018)
2. Engineering Physics, R K Gaur and S L Gupta, Dhanpat Rai Publishing Co Pvt Ltd (2015)
3. Concepts of Modern Physics, Arthur Beiser, Shobhit Mahajan and S. Rai Choudhury, McGraw Hill Education (2017)

Reference Books

1. Fundamentals of Physics, Jearl Walker, David Halliday and Robert Resnick, John Wiley and Sons (2013
2. Optics, Francis Jenkins and Harvey White, Tata Mcgraw Hill (2017)
3. Principles of Physics, John W. Jewett, Cengage publishing (2013
4. Introduction to Solid State Physics, C. Kittel, Wiley and Sons (2004)
5. Principles of Solid-State Physics, H. V. Keer, New Age International (1993)
6. Laser and Non-Linear Optics, B. B. Laud, New Age International Private Limited (2011)
7. Nanotechnology: Principles and Practices, Dr. S. K. Kulkarni, Capital Publishing Company (2014)
8. Science of Engineering Materials- C.M. Srivastava and C. Srinivasan, New Age International Pvt. Ltd. (1997)
9. Introduction to Electrodynamics –David R. Griffiths, Pearson (2013
10. Renewable Energy: Power for a Sustainable Future, Boyle, Oxford University Press (2012)

List of Assignments

Six assignments to be given by the course coordinator (Theory)-one from each unit/one mini project with report-students can work in group of 4 Maximum.

List of Laboratory Exercises

1. Study of lissajous figure by Cathode Ray Oscilloscope (CRO)
2. Determination of e/m by Thomson method
3. Determination of radius of planoconvex lens/wavelength of light/Flatness testing by Newton's rings
4. Determination of wavelength of light using diffraction grating
5. Determination of resolving power of telescope
6. Determination of thickness of a thin wire by air wedge
7. Determination of refractive index for O-ray and E-ray
8. Determination of divergence of a laser beam
9. Particle size by semiconductor laser
10. Determination of wavelength of laser by diffraction grating
11. To study Hall effect and determine the Hall voltage
12. Calculation of conductivity by four probe methods
13. Study of solar cell characteristics and calculation of fill factor
14. Determination of band gap of semiconductor
15. Determination of Planck's Constant by photoelectric effect

Project Based Learning - Provisional List of Projects

1. Measurement and effect of environmental noise in the college
2. Design and simulation of automatic solar powered time regulated water pumping
3. Solar technology: an alternative source of energy for national development
4. Design and construction of digital distance measuring instrument
5. Design and construction of automatic bell ringer
6. Design and construction of remote-control fan
7. Design and construction of sound or clap activated alarm
8. Electronic eye (Laser Security) as auto switch/security system
9. Electric power generation by road power
10. Determination of absorption coefficient of sound absorbing materials
11. Determination of velocity of O-ray and E-ray in different double refracting materials
12. Need of medium for propagation of sound wave
13. Tesla Coil
14. Thin film interference in soap film-formation of colours
15. LiFi- wireless data transfer system using light

(Note: - *Students in a group of 3 to 4 shall complete any one project from the above list)

Syllabus for Unit Tests:

Unit Test -1	Unit – I, Unit – II, Unit - III
Unit Test -2	Unit – IV, Unit – V, Unit - VI

Numerical Computation

Teaching Scheme		Examination Scheme		Credit Scheme	
	Hours/Week		Marks		Credits
Lecture:	04 Hours/Week	University Examination	60 Marks	Lecture	04
		Internal Assessment	40 Marks		
		Total	100 Marks	Total	04

Course Objective

To equip students with the knowledge of:

- Numerical methods to solve linear and system of linear equations.
- Numerical methods for differentiation and integrations.
- Numerical methods for ordinary and partial differential equations

Prerequisite:

Mathematics for Computing - I

Course Outcomes: On completion of the course, students will have the ability to:

1. Apply methods to solve linear and transcendental equations.
2. Solve system of linear equations.
3. Compute finite differences.
4. Apply method for numerical differentiation and integration.
5. Solve ordinary differential equations numerically.
6. Apply methods to solve partial differential equations.

Unit I Solution of Algebraic and Transcendental Equation **08 Hours**

Bisection method, Method of false position, Newton's method and Newton-Raphson method, Approximate solution of equation – Horner's method

Unit II Solution of Linear Simultaneous Equation **08 Hours**

Gauss elimination method, Gauss-Jordan method, Crout's triangular method, Iterative method of solution- Jacobi iteration method, Gauss-Seidal iteration method, Relaxation method

Unit III Finite Differences **08 Hours**

Forward difference operator, Backward difference operator, Central difference operator, Newton's interpolation formula, Newton's forward–backward-central interpolation formula, Sterling formula, Bessel's formula, Interpolation with unequal intervals.

Unit IV Differentiation and Integration **08 Hours**

Newton-Cortes's formula, Trapezoidal rule, Simpson one–third rule, Simpson three- eighth rule, Weddle's rule.

Unit V Numerical Solution of ODE **08 Hours**

Picard's methods, Taylor series method, Euler's method, Modified Euler's method, Runge – Kutta method, Predictor–corrector method, Milne's method. Adams-Bash fourth method, Second–order differential equation

Unit VI Finite Difference Methods **08 Hours**

Finite difference methods for solving second order two - point linear boundary value problems - Finite difference techniques for the solution of two-dimensional Laplace's and Poisson's equations on rectangular domain – One dimensional heat flow equation by explicit and implicit (Crank Nicholson) methods – One dimensional wave equation by explicit method

Textbooks

1. Burden, R.L and Faires, J.D, "Numerical Analysis", 9th Edition, Cengage Learning, 2016.
2. Grewal, B.S., and Grewal, J.S., "Numerical Methods in Engineering and Science", Khanna Publishers, 10th Edition, New Delhi, 2015.
3. S. S. Shastri, Introduction to Numerical Methods, PHI Publication.
4. V. Rajaraman, Computer Oriented Methods, 3rd Edition, PHI Publication.

Reference Books

1. Steven C Chapra, Numerical Methods for Engineers, 5th Edition, McGraw Hill Publication
2. James F. Epperson, An Introduction to Numerical Methods and Analysis, 2nd Edition, Wiley Publication.

List of Assignments

Six assignments to be given by the course coordinator (Theory)-one from each unit.

Project Based Learning - Provisional List of Projects

Students are expected prepare report on any one topic, write its definition, applications and illustrate with few examples. Also, write pseudo code for it, wherever applicable.

1. Bisection method
2. Newton Raphson's method
3. Horner's method
4. Crouts triangular mehod
5. Gauss Seidel method
6. Jacobi Method
7. Interpolation
8. Trapezoidal Rule
9. Simpson's rules
10. Euler method
11. Runge kutta method
12. Finite difference technique
13. Crank Nicolson method
14. Predictor Corrector method
15. Relaxation method

Syllabus for Unit Tests:

Unit Test -1	Unit – I, Unit – II, Unit - III
Unit Test -2	Unit – IV, Unit – V, Unit - VI

Electrical Technology

Teaching Scheme		Examination Scheme		Credit Scheme	
	Hours/Week		Marks		Credits
Lecture:	04 Hours/Week	University Examination	60 Marks	Lecture	04
Practical:	02 Hours/Week	Internal Assessment	40 Marks		
		Term work	25 Marks	Practical	01
		Total	125 Marks	Total	05

Course Objective

To equip students with the knowledge of power system basics, magnetic circuits electrical machines, transformers, wiring, measurements, illumination, and batteries.

Prerequisite:

-

Course Outcomes: On completion of the course, students will have the ability to:

1. Explain the various parameters related to magnetic circuit.
2. Describe basic concepts of AC fundamentals and circuits.
3. Illustrate constructional features and describe different parameters of transformer.
4. Describe basic concepts of power system and three phase circuits.
5. Demonstrate AC and DC electrical machines.
6. Classify types of batteries.

Unit I Magnetic Circuits **08 Hours**

Magnetic effect of electric current, Cross & Dot Convention, Right hand thumb rule, Concept of flux, flux linkages, magnetic field, magnetic field strength, magnetic field intensity, absolute permeability, relative permeability Kirchhoff's laws for magnetic circuits. Magnetic circuit concepts, analogy between electric & magnetic circuits, magnetic circuits with DC and AC excitations, magnetic leakage, B-H curve, hysteresis and eddy current losses, magnetic circuit calculations, mutual coupling.

Unit II AC Fundamentals and circuits: AC Fundamentals **08 Hours**

Sinusoidal, square, and triangular waveforms – average and effective values, form and peak factors, concept of phasor, phasor representation of sinusoidally varying voltage and current. Analysis of series, parallel and series parallel RLC Circuits: apparent, active & reactive powers, power factor, causes and problems of low power factor, power factor improvement; resonance in series and parallel circuits, bandwidth, and quality factor (simple numerical problems.

Unit III Single Phase Transformer **08 Hours**

Faradays law of electromagnetic induction, statically and dynamically induced emf, self-inductance, mutual inductance, coefficient of coupling. Single Phase Transformer: Principle of operation, construction, e.m.f. equation, voltage ratio, current ratio, KVA rating, determination of efficiency and regulation by direct load test, equivalent circuit, power losses, (simple numerical problems), introduction to auto transformer. Three phase transformer and its different winding connections.

Unit IV Introduction to Power System and Three Phase **08 Hours**

Circuits: General layout of electrical power system and functions of its elements, standard transmission and distribution voltages, concept of grid (elementary treatment only) Power generation to distribution through

overhead lines and underground cables with single line diagram. Three phase system-its necessity and advantages, meaning of phase sequence, star and delta connections, balanced supply and balanced load, line, and phase voltage/current relations, three phase power and its measurement (simple numerical problems).

Unit V Electrical Machines **08 Hours**

DC & AC: Principles of electromechanical energy conversion, DC machines: types, e. m. f. equation of generator and torque equation of motor, characteristics, and applications of dc motors (simple numerical problems). single Phase Induction motor: Principle of operation and introduction to methods of starting, applications. Three Phase Induction Motor: types, Principle of operation, slip-torque characteristics, applications (numerical problems related to slip only).

Unit VI Batteries **08 Hours**

Basic idea of primary and secondary cells, Construction, working principle and applications of Lead-Acid, Nickel Cadmium and Silver-Oxide batteries, charging methods used for lead-acid battery (accumulator), Care and maintenance of lead-acid battery, Series and parallel connections of batteries, General idea of solar cells, solar panels and their applications, Introduction to maintenance free batteries, Safe disposal of Batteries; Fuel cell: Principle & Types of fuel cell.

Textbooks

1. B.L.Theraja, A Textbook of Electrical Technology, Vol.1, S.Chand& Company Ltd. New Delhi
2. V.K.Mehta, Basic Electrical Engineering, S Chand & Company Ltd. New Delhi.
3. J.Nagarath and Kothari, Theroy and applications of Basic Electrical Engineering, Prentice Hall of India Pvt. Ltd.

Reference Books

1. Electrical Technology - Edward Huges (Pearson
2. Basic Electrical Engineering - D. P. Kothari, J Nagarath (TMC)
3. Electrical power system technology - S. W. Fordo, D. R. Patric (Prentice Hall)
4. Electrical, Electronics Measurements and Instruments - (Satya Prakashan)

List of Assignments

Six assignments to be given by the course coordinator (Theory)-one from each unit.

List of Laboratory Exercises

1. Plotting B-H characteristics for a material.
2. Load test on single phase transformer.
3. Testing and maintenance of batteries.
4. Verification of voltage and current relationships in star and delta connected 3-phase networks.
5. Load test on DC machine.
6. To find the performance of series R-L-C circuit at different condition
7. OS & SC test on single phase transformer to find efficiency and regulation
8. Speed control of DC motor
9. Study of different types of starters for DC & AC Machine
10. Load test on 3 phase Induction moto

Project Based Learning – Provisional List of Projects

1. Building a small resistive load lamp bank
2. Building a small resistive load lamp bank for various types of connections like series, parallel, star, delta
3. Building a small inductive load lamp bank for various types of connections like series, parallel, star, delta
4. Building a small capacitive load lamp bank for various types of connections like series, parallel, star, delta
5. Building a small resistive load lamp bank
6. Building a staircase wiring model on a board
7. Building a Go down wiring model on a board
8. Rewinding of a choke
9. Rewinding of a small transformer
10. Building a small rectifier circuit on bread board
11. Building a mobile charger circuit on a bread board
12. Building an electric buzzer circuit
13. Building a solar charger for mobile phone
14. Building a small wind turbine
15. Small Agricultural pump model with DC motor
16. Small Agricultural pump model with AC motor

Syllabus for Unit Tests:

Unit Test -1	Unit – I, Unit – II, Unit - III
Unit Test -2	Unit – IV, Unit – V, Unit - VI

Paradigms of Programming

Teaching Scheme		Examination Scheme		Credit Scheme	
	Hours/Week		Marks		Credits
Lecture:	04 Hours/Week	University Examination	60 Marks	Lecture	04
Practical:	02 Hours/Week	Internal Assessment	40 Marks		
		Term Work & Practical	50 Marks	Practical	01
		Total	150 Marks	Total	05

Course Objective

The course aim to make students aware of various programming paradigms and emphasising on using object-oriented approach to solve real world problems.

Prerequisite:

-

Course Outcomes: On completion of the course, students will have the ability to:

1. Demonstrate the knowledge of different programming paradigms.
2. Demonstrate the concepts of Object-Oriented Paradigm.
3. Develop programs using object-oriented approach
4. Develop small size programs using different programming language and Paradigm
5. Compare the strengths and weakness of different programming language specific to application context
6. Recognize the concepts of same kind from different programming languages and paradigms

Unit I Overview of programming paradigms **08 Hours**

Basic elements of programming languages, compiled vs. interpreted, syntax, semantics, data types, Imperative languages and non-imperative, Scripting languages, Data-oriented languages, Object-oriented languages, Event-driven Programming

Unit II Functional Programming **08 Hours**

Definition of a function: domain and range, total and partial functions, strict functions, Recursion, Referential transparency

Unit III Logic Programming **08 Hours**

Basic constructs, Facts, rules, queries, processing, goals, predicates, variables, existential queries, conjunctive queries, Definition, and semantics of a logic program.
Recursive programming: Computational model of logic programming

Unit IV Object Oriented programming **08 Hours**

Basic concepts: objects, classes, methods, overloading methods, messages inheritance: overriding methods, single inheritance, multiple.
inheritance Interfaces (e.g., in Java), encapsulation, polymorphism.

Unit V Overview of Languages **08 Hours**

Ruby: basic concepts, interpreter, strings, control structures, conditionals, loops, (duck) typing, arrays, hashes, symbols.
Prolog: structures, matching structures, equality, comparison operators, arithmetic's, lists, splitting lists, enumerating lists.
Haskell: introduction, basic concepts, basic functions, conditionals, lists, ranges, list comprehension, basics of the type system, modules

Unit VI Advanced Programming **08 Hours**

Concurrent programming, serial vs. parallel programming, process communication, basic concepts, data types, atoms, variables, pattern matching, lists, tuples- Database Programming

Erlang - Erlang: modules, functions, local functions, multiple matching possibilities, recursive functions, function evaluation, guards, control structures, anonymous functions, higher-order functions, lists and higher-order functions.

Textbooks

1. Seven Languages in Seven Weeks, Bruce A. Tate, Pragmatic Bookshelf, 2010
2. Programming Languages: Principles and Paradigms, Maurizio Gabrielli, Simone Martini, Springer, 2010
3. Programming Languages - Principles and Paradigms, Allen B. Tucker, Robert E. Noonan: (2nd ed.) McGraw-Hill, 2007
4. Clark R. G., Comparative Programming Languages, Addison-Wesley (3rd Ed.), 2000.
5. Mitchell, J. C. Concepts in Programming Languages, Cambridge University Press, 2002
6. Sebesta, R. W., Concepts of Programming Languages, Global Edition, Addison-Wesley (11th Ed.) 2016
7. Programming Languages: Concepts and Constructs; 2nd Edition, Ravi Sethi, Pearson Education Asia, 1996.

Reference Books

1. Programming Language Principles and Practice by KC Louden
2. Language manuals and on-line resources for programming languages, tools, and projects.

List of Assignments

Six assignments to be given by the course coordinator (Theory)-one from each unit

List of Laboratory Exercises

1. Write a Simple Program (as given by course coordinator) in Ruby
2. Write a simple Program (as given by course coordinator) in Prolog
3. Write a simple Program (as given by course coordinator) in Haskell
4. Write a simple Program (as given by course coordinator) in Erlang
5. Write a program to Implement Concept of Class and Objects.
6. Write a Program to Implement Concept of Method Overloading and Method Overriding
7. Write a program to implement Concept of Inheritance.
8. Write a program to implement Concept of Interface.
9. Write a program to implement Concept of Recursive Function.
10. Study of Database Programming Language approach.

Project Based Learning - Provisional List of Projects

Use the best programming paradigm for the following:

1. Operations on Matrix
2. Recursion
3. Referential transparency
4. The countdown problem
5. tic-tac-toe
6. Lazy evaluation strategy

7. Assume that you have a list of temperature readings from several cities in the world. Some of them are in Celsius and some in Fahrenheit. First let us convert them all to Celsius, then let us print the data neatly.
8. Implement a better password protection scheme: In the program {User, Password} pairs are sent in plain text over the net. Implement a scheme where the password is never stored, instead store the MD5 checksum of the password and transmit this over the net.
9. All users have the same rights: Implement a scheme whereby different users are restricted to which directories they may access.
10. Files are sent as atomic actions: Files are read, transmitted, and written as atomic actions. This may not work if the files become very large. Implement a scheme for sending the files in smaller chunks. Implement a scheme whereby an FTP transfer can be aborted and restarted in the case where we transfer very large files.

Programming Technologies and Tools Laboratory – II

Teaching Scheme		Examination Scheme		Credit Scheme	
	Hours/Week		Marks		Credits
Practical:	04 Hours/Week	Term Work & Practical	50 Marks	Practical	02
		Total	50 Marks	Total	02

Course Objective

The course focuses on making students learn and practise the Object-Oriented programming, to use concepts and solve the problems.

Prerequisite:

-

Course Outcomes: On completion of the course, students will have the ability to:

1. Choose and apply different Concepts of OOP
2. Demonstrate the use of functions to solve real world problem
3. Identify and apply the concept of Access Specifiers, Scope Resolution operator, Data Abstraction
4. Compare different types of inheritance to solve given problem.
5. Develop applications with constructor and polymorphism.
6. Develop OOP applications using file Handling and Exception handling.

Unit I Introduction to Object Oriented Programming with C++ **08 Hours**

Introduction to Object Oriented Programming, Basic Concept of OOP, Need for OOP, Benefits of OOP, Object Oriented Languages, Applications of OOP. C verses C++, C++ Characteristics, Structure of C++ program, Tokens, Keywords, Identifiers and Constants, Data Types, Declaration of variables, Dynamic initialization of variables, Control Structures

Unit II Functions in C++ **08 Hours**

The Main Function, Function Prototyping, Call by Reference, Inline functions, Default arguments, Function Overloading, Operator Overloading, Operator precedence, Math library functions, Friend and Virtual Functions.

Unit III Classes and Data Abstraction **08 Hours**

Class specification, Class Objects, Scope resolution operator, Access specifiers Public, Private, Protected, Defining member Functions, Nesting of Member Functions, Private Member Functions, Static Data Members, Static Member Functions, Data hiding.

Unit IV Inheritance **08 Hours**

Defining Derived Classes, Types of Inheritance, Virtual Base Class, Abstract class. Inheritance and protected members, protected base class inheritance, Inheriting multiple base classes, Templates: Class template, class template with parameter, function template, function template with parameter.

Unit V Constructor and destructor **08 Hours**

Types of Constructors, Types of copy constructor, constructor overloading, constructor with default parameter, dynamic initialisation of objects, destructor.

Polymorphism: Base class, Virtual Functions, Pure Virtual Functions, Calling a virtual function through a base class reference, Early and Late Binding.

Unit VI Managing, I/O and Working with Files **08 Hours**

C++ stream classes, Unformatted IO operations, formatted IO operations, Classes for file stream operations, opening and closing files, Different File Operations in C, Exception handling in C++.

Textbooks

1. E. Balaguruswamy, "Object Oriented Programming using C++", 4th Edition,Mc Graw Hill, 2010.
2. The C++ Programming Language, Bjarne Stroustrup.

Reference Books

1. Yashwant Kenetkar," Let us C++",1stEd., Oxford University Press (2006)
2. Bjrane Stroustrup, "C++ Programming language",3rd edition, Pearson education Asia (1997)

List of Laboratory Exercises

1. Explain basic concept of OOP, characteristics of OOP, Difference between C and CPP.
2. Demonstrate Basic simple CPP Program and Program related Control structures in CPP.
3. Demonstrate Concept of Function in CPP.
4. Demonstrate Concept of Inline Function in CPP.
5. Demonstrate Concept of Function Overloading and Operator Overloading in CPP.
6. Demonstrate Concept of Class and Object with the help of Scope Resolution Operator in CPP.
7. Demonstrate Concept of Different types of inheritance in CPP.
8. Demonstrate Concept of Constructor and Destructor in CPP
9. Demonstrate Concept Friend and Virtual Function in CPP
10. Demonstrate Concept of File handling and Exception handling in CPP.

Project Based Learning - Provisional List of Projects

1. Billing Application
2. Traffic Management System
3. Library Management System
4. Employee Record System
5. Security System
6. Calendar Application
7. Medical Store Management System
8. Cricket Score Sheet
9. Bank Management System
10. Telecom Billing System

Computer System Workshop Technology

Teaching Scheme		Examination Scheme		Credit Scheme	
	Hours/Week		Marks		Credits
Practical:	02 Hours/Week	Term Work & Practical	100 Marks	Practical	01
		Total	100 Marks	Total	01

Course Objective

This course focuses on enabling students to identify the hardware components of computer, assembling them, running diagnosis, carry out system configurations and installing system and user applications necessary for computing courses.

Prerequisite

-

Course Outcomes: On completion of the course, students will have the ability to:

1. Identify the architecture of a computer and its different components, including their technology evolution.
2. Apply their knowledge about computer peripherals to identify problems.
3. Install and uninstall given software step-by-step
4. Demonstrate the working of Internet
5. Prepare document using Latex
6. Use GitHub tool for coding and collaboration.

Unit I Computer Hardware **06 Hours**

Introduction to hardware components, random access memory (RAM), Types of RAM & their speed, tips for buying RAM, how to add memory to a computer, problems when installing memory, Central Processing Unit (CPU), Types Of CPU: considerations when buying a new CPU (Types & Differences), different speeds available for CPU and what do they mean, 32 Bit vs 64 Bit – Which One to Choose & Why? How to choose a CPU type for different needs? Graphic Card & Types, how to install a Graphics Card, installing a CD or DVD burner, Jumper Switch settings, Hard Disk upgrade, Different ports and why we use them - USB, PS2, DivX, Graphic card & types, Virtual Memory and how to configure it for optimum system performance.

Unit II Assembly of Computer and Software Installations **06 Hours**

Assembling the motherboard, replacing fan, how to avoid common mistakes during assembly, Installation of system software: Operating system (Windows and Linux), Installation's step for operating system, Dual booting, Configure the BIOS, Installation of Antivirus, Installation of the open-source software such as Scilab, Latex, Installation of Ms Office.

Unit III Basic Diagnostic of Hardware and Software **06 Hours**

Diagnosis of Power Up problem, Boot Drive, Errant Keyboard, mouse problems, slow computer performance, Computer freezes and displays BSOD (Blue screen of death), no display on monitor, no sound, computer rebooting or turning itself off, how to troubleshoot a computer that does not boot, Registry Cleaner.

Unit IV Computer Network Environments **06 Hours**

Network connecting devices. Configure the TCP/IP setting, connect to Local Area Network and access the Internet, Configuring Wireless network. Server and Its Configuration, Email Clients, Browsers, Office tools, customize web

browsers with the LAN proxy settings, bookmarks, search toolbars and pop-up blockers, Browsing netiquettes and cyber laws. Cloud Access Tools.

Unit V Configuration of External devices **06 Hours**

Physical set-up of Printers- Performing test print out, Printing of document etc, Scanner set-up, Webcam, Bluetooth device, Memory card reader, Connecting and Using Projectors.

Unit VI Productivity Tools **06 Hours**

Open-Source Tools Such as Latex, GitHub, Latex: Format words, lines, and paragraphs, design pages, create lists, tables, references, and figures in LATEX. Introduction to LaTex Packages and classes. Using Git, Version Control Systems, interacting with GitHub, Reverting Changes, Creating Pull Requests.

Textbooks

1. Introduction to Information Technology, ITL Education Solutions limited, Pearson Education.
2. PC Hardware and A+Handbook – Kate J. Chase PHI (Microsoft)
3. LaTeX Companion – Leslie Lamport, PHI/Pearson
4. Scilab, from theory to practice Scilab: I. Fundamentals Perrine Mathieu, Philippe Roux 2016
5. ISBN: 978-2-8227-0293-5

Reference Books

1. Computer Fundamentals, MS Office, and Internet & Web Technology by Dinesh Maidasani.
2. IT Essentials PC Hardware and Software Companion Guide Third Edition by David Anfinson and Ken Quamme. – CISCO Press, Pearson Education.
3. https://nptel.ac.in/courses/106/105/106105081/
4. http://nptel.ac.in/courses/106105084/

List of Laboratory Exercises

1. Demonstrate the Computer Hardware Components and explain its working.
2. Demonstrate the Networking Components and explain its working.
3. Installation of operating system MS windows, Unix on the personal computer
4. Installation of Application software Scilab, Latex, MS office on the personal computer
5. Troubleshooting hardware related problem.
6. Customize web browsers with the LAN proxy settings, bookmarks, search toolbars and pop-up blockers. Also, plug-ins like Macromedia Flash and JRE for applets should be configured.
7. Execution of Important "layout" and formatting commands in Latex,
8. Installation of Antivirus and customize the browsers to block pop ups, block active x downloads to avoid viruses and/or worms
9. Using Scilab commands, perform basic arithmetic and matrix operations
10. Create a Scilab script file to dis-play product of a matrix A and inverse of A.

Project Based Learning - Provisional List of Projects

1. Collect specifications of similar types of hardware and software and prepare report comparing them
2. Assembling and disassembling the PC back to working condition.
3. Installation of operating systems LINUX on Server and different packages on a PC.

4. Practice hardware troubleshooting exercises related to various components of computer like monitor, drives, memory devices, printers etc. and software troubleshooting related to BIOS etc
5. To start your own computer repair workshop. What would your initial planning involve? What would you look for in terms of building, furnishings, tools and any other equipment that you can think of?
6. Cyber Hygiene: Installing antivirus for Windows.
7. Prepare the report of need of programming language in 21st century.
8. Collect various types of computer hardware and prepare summary report
9. Prepare Seminar report using LaTex
10. Prepare Project report using LaText

B.TECH (Computer Engineering)

SEMESTER – III

COURSE SYLLABUS

Discrete Mathematics and Applications

Teaching Scheme		Examination Scheme		Credit Scheme	
	Hours/Week		Marks		Credits
Lecture:	03 Hours/Week	University Examination	60 Marks	Lecture	03
Tutorial:	01 Hours/Week	Internal Assessment	40 Marks	Tutorial	01
		Total	100 Marks	Total	04

Course Objective

The courses emphasise on mathematical foundation required for computing enabling the students to develop logical thinking, reasoning, and problem-solving skills.

Prerequisite:

Elementary Linear Algebra, Mathematics for Computing-I

Course Outcomes: On completion of the course, students will have the ability to:

1. Demonstrate the ability to write and evaluate a proof technique.
2. Apply the basic principles of set theory to analyse the data relationship and prove basic properties of set.
3. Demonstrate an understanding of relations and functions to determine their properties.
4. Apply the knowledge of Boolean algebra for building basic electronic and digital circuits.
5. Solve problems of combinatorics and recurrence relations.
6. Model problems in Computer Science using graphs and trees.

Unit I Mathematical Logic **06 Hours**

Propositional Logic, Predicate logic, First order logic, Rules of inference, Introduction to proof techniques, resolution, Mathematical induction, Methods of proofs, Applications.

Unit II Set Theory **06 Hours**

Types of sets, Sets operations and laws, Algebra of Sets, Multisets, Application of the principle of inclusion and exclusion.

Number Theory: Modular arithmetic, prime numbers, and properties, GCD, Chinese remainder theorem, Solving congruences, Applications of congruences Cryptography.

Unit III Relations **06 Hours**

Basic definition, properties and types of relations, relations and digraphs, paths in relations and digraphs, equivalence and partially ordered relations, Transitive closure and Warshall's algorithm.

Functions: Types of functions, Identity functions, Composition of functions, Mathematical functions, Pigeonhole principle.

Unit IV Algebraic Structures **06 Hours**

Isomorphism and Homomorphism Groups, Algebraic Structures with Binary Operations, rings, Cyclic groups, codes.

Lattice: Posets and Hasse Diagrams, Lattice as an algebraic system, Properties of lattices.

Group Codes: The Communication Model and Basic notion of Error Correction, Generation of Codes, Parity Checks, Error recovery in group codes.

Unit V Combinatorics and Recurrence Relations **06 Hours**

Combinatorics: Permutations, Sum rule, Product rule, Combinatorial proofs.

Recurrence Relations: Linear Recurrence relation, Second order recurrence relations with constant coefficients, Applications of Recurrence relation.

Unit VI Graph Theory and Application **06 Hours**

Definition, Degree, Types, Paths, Circuits, Operations on graphs, and Graph Models, Isomorphism, Connectedness, Planar graphs and their properties, Eulerian and Hamiltonian graphs.

Trees: Basic properties of trees, Binary trees, Application: Graph and Networks: Minimum Spanning Tree, Shortest Path, Huffman coding.

Textbooks

1. J.P. Tremblay and Manohar: Discrete mathematical structures with application to Computer Science, McGraw hill- New Delhi.
2. B. Kolman and R.C. Busby: Discrete mathematical structures for computer science Prentice Hall, New-Delhi.
3. S. Malik and M. K. Sen Discrete Mathematics, Cengage Learning India Pvt. Ltd.

Reference Books

1. Kenneth H. Rosen, Discrete Mathematics, and its applications Eighth Edition McGraw Hill Education
2. Stanat and McAlister, Discrete Mathematics for Computer Science, PHI
3. R.M. Somasundaram Discrete Mathematical Structures, Prentice Hall India Learning Private Limited

List of Assignments

1. Given a fact or a statement prove or disprove using suitable technique.
2. Write the given English language sentences represent in the Symbolic logic
3. Given the statement forms Infer the validity of the statement form
4. Draw a Hasse diagram and find chains and antichains
5. Find the number of ways for any event or given sample space.
6. Given a problem represent in a graph and compute the optimal solution
7. Given a communication network find the path between the given nodes

Project Based Learning - Provisional List of Projects

1. Discrete Mathematics in Railway Planning using graph theory and linear algebra.
2. Object transformations using linear algebra.
3. Discrete mathematics in cryptography.
4. In Google maps to determine fastest driving routes and times.
5. In image processing
6. In relation database using sets.
7. In cyber security using graph theory.
8. Shortest path between two cities using a transportation system.
9. Data compression system with the help of Huffman coding.
10. Find the shortest tour that visits each of a group of cities only once and then ends in the starting city using graphs.

Syllabus for Unit Tests:

Unit Test -1	Unit – I, Unit – II, Unit - III
Unit Test -2	Unit – IV, Unit – V, Unit - VI

Data Structures and Algorithmic Thinking

Teaching Scheme		Examination Scheme		Credit Scheme	
	Hours/Week		Marks		Credits
Lecture:	04 Hours/Week	University Examination	60 Marks	Lecture	04
Practical:	02 Hours/Week	Internal Assessment	40 Marks		
		Term Work & Practical	75 Marks	Practical	01
		Total	175 Marks	Total	05

Course Objective

The course enables students to perform tasks that facilitates them to understand interaction between the algorithms and the structure of the data being analysed by these algorithms. This course also focuses to train students in process of algorithmic thinking enabling them to build simpler solutions to various computational problems.

Prerequisite:

Classical Data Structure, Computational Thinking and Programming Concepts, Programming Technologies, and Tools Laboratory 3

Course Outcomes: On completion of the course, students will have the ability to:

1. Comprehend the real time problem.
2. Practise and apply Iterative Thinking
3. Practise and apply Recursive Thinking
4. Demonstrate the use of ADTs,
5. Develop code to illustrate sorting and searching algorithms.
6. Apply algorithms and data structures in various real-life software problems.

Unit I Recursion and Backtracking **08 Hours**

Introduction to recursion, why recursion, Format of a Recursive Function, Recursion and Memory, Recursion Vs. Iteration, Algorithms of Recursion, Recursion problems and Solutions.

Introduction to Backtracking, Algorithms of Backtracking, Backtracking problems, and Solutions.

Unit II Trees **08 Hours**

Introduction to Trees, Binary Trees, Types of Binary Trees, Properties of Binary Trees, Binary Tree Traversals, Generic Trees (N-ary Trees), Threaded Binary Tree Traversals (Stack or Queue-less Traversals), Expression Trees, XOR Trees, Binary Search Tree, Balanced Binary Search Trees, Adelson-Velskii and Landis (AVL) Trees.

Unit III Priority Queues & Heaps and Disjoint Sets ADT **08 Hours**

Introduction to Priority Queues, Priority Queue ADT, Priority Queue Applications, Priority Queue Implementations, Heaps and Binary Heaps, Heapsort, Priority Queue problems and Solutions.

Disjoint Sets ADT – Introduction, Equivalence Relations and Equivalence Classes, Disjoint Sets ADT, Trade-off in Disjoint Sets ADT implementations, Fast UNION Implementation – Slow Find and Quick Find.

Unit IV Graphs Algorithm **08 Hours**

Introduction to Graphs, Application of Graphs, Graph Representation, Graph Traversals, Topological Sort, Shortest Path Algorithms, Minimal Spanning Tree, Graph Algorithm problems & Solutions.

Unit V Sorting and Searching **08 Hours**

Introduction to Sorting, Classification of Sorting algorithms, Bubble Sort, Selection Sort, Insertion Sort, Shell Sort, Merge Sort, Heap Sort, Quick Sort, Tree Sort, Comparison of Sorting algorithms, Linear Sorting Algorithms, Counting Sort, Bucket Sort, Radix Sort, Topological Sort, External sorting.
Introduction to Searching, Types of Searching, Unordered searching, ordered linear search, Binary search, Interpolation search, Comparison of searching algorithm.

Unit VI Maps, Hash tables and Skip Lists **08 Hours**

Introduction - Hashing, Hash Tables – Hash Functions, Collision Handling schemes, Load Factors, Rehashing, and Efficiency.
Map ADT Introduction, Counting Word Frequencies, Sorted Maps, Sorted Search Tables, Skip Lists – Search and Update Operations in a skip list.

Textbooks

1. Data Structures: A Pseudo code approach with C, R. Gillberg, B. Forouzn
2. Data structures using C and C++ by Langsam, Augenstein, Tenenbaum,PHI publication
3. Data Structure and Algorithmic Thinking with Python, CareerMonk Publications, Narasimha Karumanchi, 2016

Reference Books

1. Data Structures and Algorithms in Python, Michael T. Goodrich, Roberto Tamassia, Michael H. Goldwasser, John Wiley & Sons, 2013
2. Think Data Structures- Algorithms and Information Retrieval in Java, Allen B. Downey, O'Reilly, 2017

List of Assignments

1. Write algorithm, pseudocode, and code to solve Recursion Problems like – Towers of Hanoi, whether given array is in sorted order.
2. Write algorithm, pseudocode, and code to solve Backtracking Problems like - Generate all the binary strings with n bits. Assume A [O. n - 1I is an array of size n, generate all the strings of length 11 drawn from 0... k - 1.
3. Write algorithm, pseudocode, and code to solve problems like - Give an algorithm for finding maximum element in binary tree.
4. Write algorithm, pseudocode and code to solve AVL Trees problems - Given a height h, give an algorithm for generating the HB(0). HB(0) is generating full binary tree.
5. Prove, is there a min-heap with seven distinct elements so that the pre-order traversal or it gives the elements in sorted order?
6. Travelling Salesperson Problem: Find the shortest path in a graph that visits each vertex at least once, starting and ending at the same vertex'?

List of Laboratory Exercises

1. Finding the length of connected cells of l s (regions) in a matrix of Os and ls.
2. Give an algorithm for finding the maximum clement in binary tree without recursion.
3. Give an algorithm for searching a clement in binary tree.
4. Give an algorithm for finding the diameter of the binary tree. The diameter of a tree (sometimes called the width) is the number of nodes on the longest path between two leaves in the tree.
5. Implement the Algorithm for Building Expression Tree from Postfix Expression
6. Write and implement an algorithm for deleting an arbitrary clement from min heap.

7. Write and implement an algorithm for checking whether a given graph G has simple path from source s to destination d. Assume the graph G is represented using the adjacent matrix.
8. Perform DFS on given graph C.
9. Count the number of connected components of Graph G which is represented in the adjacent matrix.
10. Merging K sorted lists: Given K sorted lists with a total of n elements, write an algorithm to produce a sorted list of all n elements.

Project Based Learning - Provisional List of Projects

1. Design and development of Student attendance system using array data structure.
2. Design and development of Car rental system using Singly linked list (SSL) data structure.
3. Design and development of Inventory management system using suitable data structure.
4. Comparative study of student management system using array, queue, and stack.
5. Design phone dictionary using doubly linked list (DLL).
6. Design and implement of dictionary using hierarchical data structure.
7. Design and implement of expression solver using stack.
8. Design and development quizer (quiz conduction application).
9. Design and development of subject recommendation system.
10. Design and development of Sudoku Solver.

Syllabus for Unit Tests:

Unit Test -1	Unit – I, Unit – II, Unit - III
Unit Test -2	Unit – IV, Unit – V, Unit - VI

Computer Organisation and Design

Teaching Scheme		Examination Scheme		Credit Scheme	
	Hours/Week		Marks		Credits
Lecture:	04 Hours/Week	University Examination	60 Marks	Lecture	04
Practical:	02 Hours/Week	Internal Assessment	40 Marks		
		Term Work & Practical	50 Marks	Practical	01
		Total	150 Marks	Total	05

Course Objective

This course aims at providing comprehensive understanding of the organization and architecture of modern-day computers, emphasizing both fundamental principles and role of performance parameters in driving computer design.

Prerequisite:

Digital Electronics

Course Outcomes: On completion of the course, students will have the ability to:

1. Analyse the design issues in terms of speed, technology, cost, performance
2. Understand the architecture and functionality of central processing unit.
3. Learn design approaches implementing control unit
4. Discuss the concept of memory organization.
5. Describe structure and functions of I/o module and Peripherals.
6. Infer Performance Enhancement of Processor

Unit I Computer arithmetic and performance **08 Hours**

Computer organization and system architecture, Structure and functions, Von Neumann Architecture, IAS machine. Computer Performance Measurement, Aspects & Factors affecting Computer Performance, MIPS & MFLOPS, designing for performance, fixed and floating-point representations, IEEE 754 format. Booths Algorithm for Signed Multiplication, Restoring and Non-Restoring Division Algorithms.

Unit II Central Processing Unit **08 Hours**

Introduction to x86 microprocessor, Architecture, register organization, Segmentation, Instruction execution cycle, addressing modes, and Instruction set. Instruction Formats, Instruction Types, the Instruction Cycle, and Instruction Pipelining, RISC Vs. CISC Architecture

Unit III Control Unit **08 Hours**

Instruction Cycle & Micro Operations, Functional Requirements & Operations of the Control Unit, Block Schematic & Control Signals, Single Bus Processor Organization, Control Signal example with Micro Operations and Register Transfer. Control Unit Design Methods – Hardwired Control and Micro-Programmed Control Microinstructions & Formats, Control Memory, Microinstruction Sequencing, Sequencing Techniques, Address Generation, Microinstruction Execution, Microinstruction Encoding

Unit IV Memory **08 Hours**

Characteristics of Memory Systems, Internal and External Memory Types. Memory Hierarchy, Principle of Locality, Cache Memory – Basics, Performance Metrics & Improvements, Organization and Mapping Techniques, Handling Cache Misses & Writes, Replacement Algorithms, Cache Controllers

Unit V I/O organization **08 Hours**

Structure and functions of I/o Module, Peripheral devices and their characteristics, Input-output subsystems, I/O device interface, Programmed I/O, Interrupt driven I/O, DMA, Buses-SCSI, USB

Unit VI Performance enhancement of processors **08 Hours**

CPU Performance and its Factors, Evaluating Performance, Enhancing Performance - Pipeline Processing, instruction pipelining, pipeline stages and hazards, The ARM Cortex–A8 and Intel Core i7 Pipelines, Parallel Processing Concepts - Flynn's classifications, Cache coherence in multiprocessor systems, Specialized Architectures - Multi-core systems

Textbooks

1. William Stallings. "Computer organization and architecture: designing for performance". Pearson Education India, 2010
2. Carl Hamacher, Zvonko Vranesic and Safwat Zaky. "Computer Organization", McGraw Hill, 2011.
3. Computer System Architecture M. M. Mano: 3rd ed., Prentice Hall of India, New Delhi, 1993.
4. Computer Architecture and Organization, John P. Hayes.

Reference Books

1. A. S. Tanenbaum "Structured Computer Organization", 4th Edition, Prentice Hall of India, 1991 ISBN: 81–203–1553–7.
2. Computer Organization and Design: The Hardware/Software Interface, David A. Patterson and John L. Hennessy.

List of Assignments

1. Describe structure of IAS computer with neat block diagram.
2. Describe architecture of 8086 with neat block diagram.
3. Draw and Explain Hardware implementation of Booth's algorithm for signed number multiplication
4. Draw flowchart for Booth's Algorithm for multiplication and perform 9 x -6
5. Draw and Explain Hardwired control unit using a. Delay element method or b. Sequence Counter method
6. Explain Direct mapping and set associative mapping of cache memory in detail with suitable example

List of Laboratory Exercises

1. Write an Assembly Language program to display system time on screen.
2. Write an Assembly Language program to add and subtract two 8-bit numbers.
3. Write an Assembly Language program to determine 2's complement of a number
4. Write an assembly language program for BCD addition and subtraction.
5. To Apply Booth's Algorithm for illustrating multiplication of signed numbers.
6. To design and illustrate Restoring Division Algorithm.
7. To design and illustrate Non-Restoring Division Algorithm.
8. To design 4-bit ALU (VLAB)
9. Study of Memory Design (VLAB)
10. To design Direct mapped Cache and associative cache (VLAB)

Project Based Learning - Provisional List of Projects

Write program to generate assembly code from prefix code.

1. Simulate a word multiplier.
2. Simulate a word divider.

3. Suggest a high-speed addition method and logic for 4-bit addition
4. Design and implement an arbitrary precision four function calculator.
5. Simulate modern traffic control system.
6. Suggest and design a minimal cpu architecture for controlling the washing machine.
7. Write/create/research a tool for benchmarking of a hardware (CPU).
8. Implement quick sort using assembly language.
9. Implement binary search using assembly language.
10. Implement matrix multiplication using assembly language
11. Microprocessor based automatic attendance recorder
12. Microprocessor based furnace temperature controller.

Syllabus for Unit Tests:

Unit Test -1	Unit – I, Unit – II, Unit - III
Unit Test -2	Unit – IV, Unit – V, Unit - VI

Computer Networks

Teaching Scheme		Examination Scheme		Credit Scheme	
	Hours/Week		Marks		Credits
Lecture:	03 Hours/Week	University Examination	60 Marks	Lecture	03
Practical:	02 Hours/Week	Internal Assessment	40 Marks		
		Term Work & Practical	75 Marks	Practical	01
		Total	175 Marks	Total	04

Course Objective

This Course emphasis on all theoretical concepts and practical aspects of networking. This course enables the students to understand the networking hardware & concepts through using network simulators.

Prerequisite

Computer System Workshop Technology.

Course Outcomes: On completion of the course, students will have the ability to:

1. Demonstrate the knowledge of computer networking.
2. Elucidate detailed structure of physical and data link layers
3. Demonstrate the knowledge of about Network and Transport Layers.
4. Elucidate the details of Session and Presentation Layers
5. Illustrate the functionality of Application layer
6. Recite the fundamentals of wireless network.

Unit I Fundamentals of Networking **06 Hours**

History of network and internet, need of network, Types of networks, Networking hardware, Information transmission, Transmitter, Receiver, Introduction to networking applications and simulators, Introduction to OSI reference model, Introduction to TCP/IP model, Introduction to internet and working of Internet, Introduction to network security, IEEE standards for networking

Unit II Physical and Data Link Layers **06 Hours**

Ethernet, LLC, MAC, Framing, Channel Allocation schemes, Error Control, Flow Control, Error Detection and correction, Physical Layer protocols, Data Link layer protocols

Unit III Network and Transport Layers **06 Hours**

Network Layer: Network Packet structure and formation, routing algorithms, congestion control algorithms, quality of service, IP Addressing, Subnets, configuring network settings, Network problem solving.

Transport Layers: Segmentation, Congestion control, Connection oriented and connection less services, Network and Transport Layer Protocols.

Unit IV Session and Presentation Layers **06 Hours**

Session Layer: Session management, synchronization, Dialog control, Presentation Layer: Encryption-decryption, Compression, File formats, Translation, Session and Presentation Layer protocols, Session and Presentation Layer protocols.

Unit V Application Layer **06 Hours**

DNS, URL, Data Cache and streaming, Web Applications, Web browser working, Cloud services, User interface and User interaction, Mail systems, Support of file formats, Application Layer protocols

Unit VI Network Security **06 Hours**

Firewall, Types of Firewalls, Cryptography, Symmetric Key Algorithm, Public Key Algorithm, Digital Signatures, Public Key Management, Communication Security, Authentication protocols.

Textbooks

1. Data and computer communications, William Stallings, 10th edition, Pearson
2. Computer networking: a top-down approach, James f. Kurose, Keith w. Ross, 6th edition, Pearson.
3. Computer Networks, Tanenbaum, 5th Edition, Pearson

Reference Books

1. Data communication & networking, Forouzan, 5th edition, McGraw-Hill.
2. Computer Networking Beginners Guide, Russell Scott, 1st edition, Stefano Cardinale

List of Assignments

1. Explain in detail the types of Networks.
2. Consider the real time scenario to explain the error correction and detection.
3. Compare: Connection oriented and Connection less services
4. Consider the real time scenario to explain the role of Session and Presentation Layer in networking
5. Explain in detail: How does the web browser work?
6. Explain in detail: The role of Network Security in computing.

List of Laboratory Exercises

1. Introduction to Computer Network and Network Simulators.
   Networking devices, Addresses, Network Security, Internet working, Network Simulators: Cisco Packet Tracer, Netemul, NetSim.
2. Network configuration of PCs and other networking devices
   Configuring Computer and Router via OS UI and commands, Networking commands
3. Network configuration of PCs and other networking devices using network simulators.
   Observing and configuring PCs, Routers, Switch, Hub, and other networking devices using network simulators
4. Establishment of simple LAN network using real time devices and network simulators.
   Establishment of simple LAN network using actual devices like PCs, Switch, Router and through network simulators
5. Communication between two or multiple systems using network simulators.
   Establishment of network where two network systems can communicate with each other. Use of PCs, Switch etc. in network simulator.
6. Broadcasting using network simulators.
   Establishing network to broadcast the information using network simulator. Use of PCs, Switch and Hub in the network simulator.
7. Establishment of different networks and communication between them using real time devices and network simulators.
   Establishment of different networks and communication between using actual devices like PCs, Switch, Router and through network simulators
8. Understanding Protocols of Transport Layer using Network Simulators
   Understanding Transport Layer protocols TCP, UDP using networking simulators
9. Implementation of Client-Server Architecture for Same Network using Network Simulators and Programming

Implementing some services of Server with protocols like FTP, HTTP, and others. Use of PCs, Servers, Switch in Network Simulator. Socket Programming: Use of Python, C, C++, Java for implementing Client Server architecture.

10. Implementation of Client-Server Architecture for different Networks using Network Simulators.
    Implementing some services of Server with protocols like FTP, HTTP, and others for different networks. Use of PCs, Servers, Switch and Router in Network Simulator.
11. Understanding Web Browser structure and working
    Web browser structure, source code loading, UI generation, Information loading into application layer, file format support and Complete working of web browser.
12. Establishment of wireless network
    Establishment of wireless networking using actual devices and via network simulator. Use of Laptops and Wifi Router.

Project Based Learning - Provisional List of Projects

1. IP based patient monitoring system
2. Configuring Internet Router
3. Configuring Network Switch
4. Home Automation system using Wi-Fi
5. Wireless Weather monitoring system using Raspberry pi.
6. Smart Traffic control system
7. Smart energy meter for homes
8. Analysis of IPv4/IPv6 protocols
9. Web System Security.
10. Personalized Web Search with Location Preferences

Syllabus for Unit Tests:

Unit Test -1	Unit – I, Unit – II, Unit - III
Unit Test -2	Unit – IV, Unit – V, Unit - VI

Software Engineering

Teaching Scheme		Examination Scheme		Credit Scheme	
	Hours/Week		Marks		Credits
Lecture:	04 Hours/Week	University Examination	60 Marks	Lecture	04
Practical:	04 Hours/Week	Internal Assessment	40 Marks		
		Term Work & Oral	50 Marks	Practical	02
		Total	150 Marks	Total	06

Course Objective:

The course emphasises on the building blocks, importance and need of software engineering. It focuses on the various processes, methods, and practices for developing software which makes software engineering as a specific discipline. This course includes elaboration on each phase of software development life cycle methodologies and practices. It further covers the essential knowledge required to ensure the quality and maintenance of developing and developed software.

Prerequisite:

-

Course Outcomes: On completion of the course, students will have the ability to:

1. Apply the Software Engineering approach to software design and development.
2. Apply the Essential processes of SDLC.
3. Demonstrate the knowledge of requirement elicitation by classifying and documenting the requirements
4. Demonstrate software design by modelling artifacts for gathered requirements & analysis.
5. Apply testing strategies and create test cases and test suites
6. Use the project management concepts and tools for managing software project.

Unit I Overview of Software Engineering & Its Methodologies **08 Hours**

Defining Software Engineering, Software Engineering Principles, Software Engineering Ethics, Software Process, Project, Product and People.

Overview of Software development lifecycle methodologies: Waterfall, Agile, Lean, Iterative, Spiral, DevOps.

Unit II Requirements and Design **08 Hours**

Expressing Requirements, Types of Requirements, Feasibility Study, Elicitation Techniques. Requirements Analysis - Structured Analysis, Object Oriented Modelling, Other Approaches. Requirement Specification, Requirement Validation, Requirement Engineering Tools (CASE).

Software Design: Principles of Software Design, Data Design, Architectural Design, Component Level Design, Object-oriented design, Design Notations, User Interface Design

Unit III Coding and Testing **08 Hours**

Coding Guidelines, Coding Methodology, Programming Practice - Top-down, bottom-up, structured programming, information hiding programming, Code verification Techniques, Introduction to No-Code Development approach and tools.

Testing: Software test Characteristics, Test plan, Test Case Design, Testing Strategies, Testing Techniques, Debugging Process, and strategies.

Unit IV Software Quality and Maintenance **08 Hours**

Quality Concepts, ISO 9126 Quality Factors, Mc Call's Quality Factors, SQA plan, SQA Activities, Software Reviews, Sig Sigma & ISO 9000 Quality Standards, capability maturity model, Software Reliability.

Maintenance: Factors affecting software Maintenance, Types of software Maintenance, Software Maintenance Lifecycle.

Unit V SCM and Re-engineering **08 Hours**

Software Configuration Management - Basics, SCM Planning, Project Library, SCM Process - Configuration Identification, Change Control, Version Control, SCM Tools (CASE).

Software re-engineering: Objectives, Principles of Re-engineering, Levels of Abstraction, Software Re-engineering process Model, Business Process Re-Engineering.

Unit VI Software Planning and Cost Estimation **08 Hours**

Responsibility of Software Project Manager, Project Planning, Project Scheduling, People capability maturity model, Risk Management.

Cost Estimation - Basics, Estimation of Resources, Product Cost Factors, Cost Estimation Process, Constructive Cost Model, Function Point Analysis, Decomposition techniques- Problem based Estimation, Process based estimation, use case-based estimation.

Textbooks

1. Fundamentals Of Software Engineering, Rajib Mall Phi Learning, 02-Apr-2014, Isbn 8120348982, 9788120348981
2. "Software Engineering: Principles and Practices, 2nd Edition by Rohit Khurana, Khurana Rohit ·2010, Vikas Publishing House Pvt Limited", ISBN: 9788125939467
3. Software Engineering principles and practices, Rajesh Narang, 2015 McGraw Hill Education, ASINB014ULF4R8

Reference Books

1. Software Engineering: A Practitioner's Approach, By Roger Pressman and Bruce Maxim, McGraw Hill, 9th Edition, ISBN10: 1259872971
2. Software Engineering, by Ian Sommerville, Pearson; 10th edition, ISBN-10: 0133943038

List of Assignments

1. A mini project will be given to the students based on which they need to prepare the following
    a. Choosing the appropriate SDLC method to develop the given project.
    b. Develop the project plan along with feasibility study and estimations
    c. Prepare the Software requirement Specification document
    d. Prepare the Software Detailed Design Document
    e. Prepare test cases
    f. Use CASE tools to perform all the above tasks.

List of Laboratory Exercises

1. Present a Case study on Agile methodology
2. Present a Case study on DevOps
3. For the given project, perform requirement elicitation using tools and prepare SRS.
4. Prepare case study on Tools used to create Unified Modelling Language.
5. Design Structure Model for the given SRS using UML tool.
6. Design Behaviour model for the given SRS using UML tool

7. Prepare the Testcases using the Junit.
8. Prepare case study on Tools used to prepare project Plan.
9. Prepare a case study on Automated testing Tools.
10. Prepare a case study on Total quality management.

Project Based Learning - Provisional List of Projects

1. Flight Vehicle and Aircraft Systems Engineering.
2. Skyscraper
3. Software piracy protection system
4. e-Learning platform
5. Bug tracker
6. Railway tracking and arrival time prediction system
7. Employee management system
8. Camera motion sensor system
9. Operating System task monitoring application
10. Data leakage detection system

Syllabus for Unit Tests:

Unit Test -1	Unit – I, Unit – II, Unit - III
Unit Test -2	Unit – IV, Unit – V, Unit - VI

Programming Technologies and Tools Laboratory – III

Teaching Scheme		Examination Scheme		Credit Scheme	
	Hours/Week		Marks		Credits
Practical:	04 Hours/Week	Term Work & Practical	50 Marks	Practical	02
		Total	50 Marks	Total	02

Course Objective:
The course aims to make students aware of python programming.

Prerequisite:
-

Course Outcomes: On completion of the course, students will have the ability to:

1. Demonstrate the knowledge of using data structures in python.
2. Demonstrate the characteristics of object-oriented Python
3. Perform basic operations on file.
4. Understand and implement error and exception handling
5. Design basic GUI using Python Tkinter
6. Implement database connectivity using MySql and SQLite.

Unit I Python Language Basics **08 Hours**

Python Interpreter, Running IPython Shell, Running Jupyter Notebook, Tab Completion, Introspection, The %run Command, Executing Code from the Clipboard, Terminal Keyboard Shortcuts, About Magic Commands, Language Semantics, Scalar types, Control flow.
Data Structures and Sequences – Tuple, List, built in sequence functions, dict, set, strings.

Unit II Functions, Modules, Packages **08 Hours**

Functions – def statement, returning values, parameters, arguments, local variables, global variables and global statement, Doc Strings, Decorators, lambda, iterators and generators, Modules, Packages.

Unit III Object-Oriented Approach **08 Hours**

Classes - A simple class, defining methods, the constructor, Member variables, calling methods, Adding inheritance, Class variables, Class methods and static methods, Properties, Interfaces, New style classes, Doc strings for classes, Private members.
Scope & Namespaces, object, instantiations, Inheritance, Multiple inheritance, Constructors, operator overloading.

Unit IV File Handling and Error **08 Hours**

Python file handling: File handling modes, Text & Binary Files – Reading, Writing and Delete.
Error and exception handling: Exceptions, Handling Exceptions, Raising Exceptions, Exception Chaining, User-defined Exceptions, Defining Clean-up Actions, Predefined Clean-up Actions.

Unit V Python Tkinter **08 Hours**

Event Driven Programming, GUI frameworks-Tkinter, Windows and windows attribute, Component, Tk widgets-Ttk and Tix widgets, Geometry Management, Events & Binding Functions, simple GUI applications development.

PyGtk – Simple message box, text input dialog box, file selection dialog box.
EasyGUI - Example

Unit VI Backend and Database **08 Hours**

Tornado for windows, building python HTTP web server (GET method), Parameter and Python API (resource and query), Building a JSON GET & POST API, the GET and POST API from JavaScript/HTML
SQLite, MySQL -Environment Setup, Database Connection, CRUD operations.

Textbooks

1. Python 3 Object-oriented Programming Second Edition, Dusty Phillips, Packt Publishing
2. MySQL for Python: Database Access Made Easy,
3. Python GUI Programming with Tkinter, Alan D. Moore, O'Reilly Media, Inc.

Reference Books

1. Introduction to Computation and Programming Using Python, John V Guttag, Prentice Hall of India
2. Python Essential Reference 4th Edition, David Beazley, Pearson Education.

List of Laboratory Exercises

1. Study about Anaconda python software.
2. Write a program to understand the control structures of python
3. Write a program to learn different types of structures (list, dictionary, tuples) in python
4. Write a program to learn concept of functions scoping, recursion, and list mutability.
5. Write a program to understand working of exception handling and assertions.
6. Write a program to perform basic operations on text files.
7. Write a program to implement HTTP server using Python
8. Write a program to implement basic GUI application with database connectivity using SQLite
9. Write a program to learn GUI programming using Tkinter
10. Write a program to implement basic GUI application with database connectivity using MySql

Project Based Learning - Provisional List of Projects

1. Design and development of Mad Libs generator.
2. Design and development of electronic mail system (Read, write, send and delete operations).
3. Design and development of store billing system.
4. Design and development of typing speed check web application.
5. Design and development of windows application for music player.
6. Design and development of windows Quiz Application.
7. Design and development of web application for daily expense tracker.
8. Design and development of student portfolio management & CV generator system.
9. Design and development of windows based to do list or sticky notes.
10. Design and development of assignment plagiarism checker

B.TECH (Computer Engineering)

SEMESTER – IV

COURSE SYLLABUS

Probability and Statistics

Teaching Scheme		Examination Scheme		Credit Scheme	
	Hours/Week		Marks		Credits
Lecture:	03 Hours/Week	University Examination	60 Marks	Lecture	03
		Internal Assessment	40 Marks		
		Total	100 Marks	Total	03

Course Objectives:

To equip students with the knowledge of:

- Probability theory and expected value.
- Probability distribution and its applications.
- Multiple regression and ANOVA.

Prerequisite:

Elementary Mathematics

Course Outcomes: On completion of the course, students will have the ability to:

1. Apply Bayes theorem to find probability.
2. Compute mathematical expectations.
3. Identify various theoretical distributions.
4. Use correlation coefficient to interpret numerical data.
5. Use regression to estimate the dependent variable.
6. Apply concept of graph in optimization.

Unit I Probability Theory **06 Hours**

Definition of probability: classical, empirical, and axiomatic approach of probability, Addition theorem of probability, Multiplication theorem of probability, Bayes' theorem of inverse probability, Properties of probabilities

Unit II Random Variable and Mathematical Expectation. **06 Hours**

Definition of random variables, Probability distributions, Probability mass function, Probability density function, Mathematical expectation, Join and marginal probability distributions, Properties of expectation and variance with proofs, Examples

Unit III Theoretical Probability Distributions **06 Hours**

Binomial distribution, Poisson distribution, Normal distribution, Fitting of binomial distributions, Properties of binomial, Poisson and normal distributions, Relation between binomial and normal distributions, Relation between Poisson and normal distributions, Importance of normal distribution

Unit IV Correlation **06 Hours**

Introduction, Types of correlation, Correlation and causation, Methods of studying correlation, Karl Pearson's correlation coefficient, Spearman's rank correlation, Coefficient, Properties of Karl Pearson's correlation coefficient, Properties of Spearman's rank correlation coefficient, Probable errors, Examples

Unit V Linear Regression Analysis **06 Hours**

Introduction, Linear and non-linear regression, Lines of regression, Derivation of regression lines of y on x and x on y, Angle between the regression lines, Coefficients of regression, Theorems on regression coefficient, Properties of regression coefficient

Unit VI Multiple Regression and AVOVA **06 Hours**

Multiple regression & multiple correlation, Analysis of variance (one way, two ways with as well as without interaction)

Textbooks

1. S. C. Gupta, "Fundamentals of Statistics", 46th Edition, Himalaya Publishing House.
2. G. V. Kumbhojkar, "Probability and Random Processes", 14th Edition, C. Jamnadas and co.
3. Murray Spiegel, John Schiller, R. ALU Srinivasan, Probability and Statistics, Schaum's Outlines
4. Kishor S. Trivedi, "Probability, Statistics with Reliability, Queuing and Computer Science Applications", 2nd Edition, Wiley India Pvt. Ltd.

Reference Books

1. Vijay K. Rohatgi, A. K. Md. Ehsanes Saleh, An Introduction to Probability and Statistics, 3 rd Edition, Wiley Publication
2. I.R. Miller, J.E. Freund, and R. Johnson.Fun "Probability and Statistics for Engineers" (4th Edition)

List of Assignments

Six assignments to be given by the course coordinator (Theory)-one from each unit.

Project Based Learning - Provisional List of Projects

Students are expected prepare report on any one topic, write its definition, applications and analyse the hypothetical data. Also, write pseudo code for it, wherever applicable.

1. Bayes theorem
2. Additive and multiplicative law of probability
3. Mathematical expectation
4. Joint and marginal probability distribution
5. Theoretical probability distribution
6. Coefficient of correlation
7. Regression estimates
8. Simple regression model
9. Multiple regression model
10. One way ANOVA
11. Two-way ANOVA
12. Correlation
13. Multiple correlation

Note: - *Students in a group of 3 to 4 shall complete any one project from the above list)

Syllabus for Unit Tests:

Unit Test -1	Unit – I, Unit – II, Unit - III
Unit Test -2	Unit – IV, Unit – V, Unit - VI

Models of Computation

Teaching Scheme		Examination Scheme		Credit Scheme	
	Hours/Week		Marks		Credits
Lecture:	03 Hours/Week	University Examination:	60 Marks	Lecture	03
Tutorial:	01 Hours/Week	Internal Assessment:	40 Marks	Tutorial	01
		Total	100 Marks	Total	04

Course Objective:
This course enables students to understand any problem by developing abstract models of computing machines and reasoning about their compute efficiency.

Prerequisite:
Discrete Mathematics, Data Structure and Algorithmic Thinking

Course Outcomes: On completion of the course, students will have the ability to:

1. Define and describe formal models of computation
2. Illustrate examples of languages and computational problems appropriate to different models of computation.
3. Demonstrate the relationships between language classes and regular expression.
4. Design grammars and recognizers for different formal languages
5. Design Language Acceptability by Turing Machine
6. Use models of computation to understand the compiler basics

Unit I Basic Concepts and Automata Theory **06 Hours**

Introduction to Theory of Computation- Automata, Alphabet, Symbol, String, Formal Languages, Deterministic Finite Automaton (DFA)- Definition, Representation, Acceptability of a String and Language, Non-Deterministic Finite Automaton (NFA), Equivalence of DFA and NFA, NFA with ε-Transition, Equivalence of NFA's with and without ε-Transition, Finite Automata with output- Moore machine, Mealy Machine, Equivalence of Moore and Mealy Machine, Minimization of Finite Automata

Unit II Regular Languages **06 Hours**

Definition and Examples. Conversion of RE to FA, FA to RE, algebraic laws, applications of RE. Pumping lemma for regular languages and applications. Closure properties of regular Languages Union, Concatenation, Complement, Intersection and Kleene closure. Decidability- Decision properties

Unit III Context Free Grammar (CFG) **06 Hours**

Definition, Derivations, Languages, Derivation Trees and Ambiguity, Regular Grammars-Right Linear and Left Linear grammars, Conversion of FA into CFG and Regular grammar into FA, Simplification of CFG, Normal Forms-Chomsky Normal Form (CNF), Greibach Normal Form (GNF), Chomsky Hierarchy, Programming problems based on the properties of CFGs

Unit IV Push Down Automata (PDA) **06 Hours**

Introduction, Pushdown Automata (PDA), Transition Diagrams, Functions and Tables, Deterministic Push-down Automata (DPDA) - definition, Nondeterministic Pushdown Automata (NPDA), Equivalence of context free grammars and PDA, properties of context free languages. Introduction to Post Machines (PMs).

Unit V Turing Machine Model **06 Hours**

The Turing Machine Model and Definition of TM, Language Acceptability of Turing Machines, Techniques for Turing Machine Construction, Modifications of Turing Machine, Universal Turing machine, Linear Bounded Automata, Church's Thesis, Halting Problem

Unit VI Basics of Compiler **06 Hours**

Introduction to Natural language Processing, Syntax analysis language definition. Primitive recursive functions – Recursive and recursively enumerable languages – Universal Turing machine. Lexical analyser, Text editor, and searching using RE.

Textbooks

1. Theory Computation, Vivek Kulkarni, Oxford higher education
2. Theory of Computer Science (Automata, Language & Computation) K. L. P. Mishra & N. Chandrasekaran, PHI Second Edition
3. Theory of Computer Science, E.V. Krishnamurthy, EWP Publication
4. Introduction to languages and the theory of computation by Jhon C Martin Mc Graw Hill
5. Introduction to Automata Theory, Languages, and Computation (third edition), by John Hopcroft, Rajeev Motwani, Jeffrey Ullman, Addison Wesley, 2007.

Reference Books

1. Introduction to Automata Theory, Hopcroft Ullman, Languages & Computations, Narosa
2. Introduction to Computer Theory, Daniel A. Cohen, Wiley Publication
3. Theory of Computation, Dexter C. Kozen, Springer Science & Business Media, 2006

List of Assignments

1. Study of JFLAP tool for Constructing FA
2. Construct regular expressions defined over the alphabet Σ= {a, b}, which denote the given languages.
3. Translate the following Mealy machine into its equivalent Moore machine.
4. Write a context-free grammar (CFG) which generates the language L denoted by:(a+ b)* bbb(a+ b) *
5. Construct a PDA that accepts the language defined by the following regular grammar.
6. Design a TM to recognize an arbitrary string divisible by 4, from Σ = {0, 1, 2}.

Project Based Learning - Provisional List of Projects

1. Develop a tool to illustrate the algorithm for converting an arbitrary NFA to a DFA.
2. Develop a tool to draw a transition diagram for any given DFA.
3. Approximation algorithms
4. Greedy algorithms.
5. Enumeration of finite automata
6. Enumeration of PDA
7. Enumeration of Turing machines
8. Ambiguous grammars
9. Disambiguation of Grammars
10. Enumeration of Context-free languages
11. Enumeration of Turing machines
12. Universal Turing machines.
13. Randomized Turing machines
14. NP Complete Algorithm

15. Problem solvability using Reduction
16. Design of TM to emulate a finite automaton
17. Design of TM to emulate a PDA
18. Complexity analysis of encryption algorithms using TM.
19. Design of TM to perform sorting
20. Design TM to perform searching.

Syllabus for Unit Tests

Unit Test -1	Unit – I, Unit – II, Unit - III
Unit Test -2	Unit – IV, Unit – V, Unit - VI

Computer Operating System

Teaching Scheme		Examination Scheme		Credit Scheme	
	Hours/Week		Marks		Credits
Lecture:	03 Hours/Week	University Examination:	60 Marks		
Practical:	02 Hours/Week	Internal Assessment:	40 Marks	Lecture	03
		Term Work & Practical	50 Marks	Practical	01
		Total	150 Marks	Total	04

Course Objective;

The course focuses on the concepts of operating systems enabling students to understand and apply the principles, structure and functioning of Operating system.

Prerequisite:

Computer architecture and Design, Data structures and algorithms, Programming Skills.

Course Outcomes: On completion of the course, students will have the ability to:

1. To learn and apply the Concepts of operating system
2. Infer the concept of process, thread and Inter process communication
3. Outline the concept of concurrency and deadlocks.
4. Analyse of Memory Management and Virtual Memory
5. Utilize the concepts of I/O System for communication
6. Illustrate the Issues in real time operating system.

Unit I Functionalities & Services of an Operating System **06 Hours**

Architecture of OS, Goals & Structures of O.S, Basic functions, System Calls & Types, Process Concept, Process Control Block, Linux System calls for Process creation, Inter Process Communication using Shared memory / Message passing.

Unit II Concurrency, Multithreaded programming **06 Hours**

Benefits, challenges, models, Pthreads library in Linux: thread creation, cancellation, thread specific data, Thread pools, Signal handling, Scheduling: Pre-emptive, non-pre-emptive algorithms FCFS, SJF, SRT, RR, Thread scheduling: contention scope, Pthread support for scheduling

Unit III Deadlock **06 Hours**

Principles of deadlock, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, System calls like signal, Producer Consumer problem (multi-threaded) example Deadlock characterization, Resource graph, Avoidance & Prevention, Safe state, Banker's algorithm, recovery schemes

Unit IV Storage Management **06 Hours**

Memory management, logical v/s physical address space, Segmentation, Paging, Page table structures, Virtual memory, Page replacement strategies, File Systems, file operations, types, access methods, Directory structure, Mounting file systems.

Unit V I/O Systems **06 Hours**

File concept, Access methods, Directory structure, Filesystem mounting, Protection, Directory implementation, Allocation methods, Free-space management, Disk scheduling, Disk management, Swap-space management, Protection.

Unit VI Issues, Protection and Security **06 Hours**

Features of real-time kernels. Real-time CPU scheduling and real-time performance issues, Goals of protection; Domain of protection; Protection models; Security, problems, and threats; Authentication; and Encryption.

Textbooks

1. Operating System Concepts, 9th edition Peter B. Galvin, Greg Gagne, Abraham Silberschatz, John Wiley & Sons, Inc.

Reference Books

1. Modern Operating Systems -By Andrew S. Tanenbaum (PHI)
2. Operating Systems 5th Edition, William Stallings, Pearson Education India

List of Assignments

1. Write in detail about the Quality (Computer Architecture) based on Features and Functionality of latest OS.
2. Discuss in detail the Concurrency mechanism and Multithreaded programming achieved in latest any OS.
3. Explain the mechanism of process and processor management in Unix/Linux OS
4. Discuss in detail the mechanism used for memory management in Linux OS
5. Elaborate in detail the user interface concepts of Linux OS.
6. Write in detail about the Quality (Security Threats) based on Features and functionality of latest OS

List of Laboratory Exercises

1. Basic functionalities and functions of operating system.
2. Write Shell Script to copy the file system from two directories to a new directory in such a way that only the latest file is copied in case there are common files in both the directories.
3. Implementation of FCFS (First Come First Serve) CPU Scheduling.
4. Implementation of SJF (Shortest Job First) CPU Scheduling.
5. Implementation of Round Robin (RR) CPU Scheduling.
6. Producer Consumer Problem Using Semaphores
7. Bankers Algorithm for Deadlock Avoidance
8. Algorithm for Deadlock Detection
9. Page Replacement Algorithms FIFO and LRU
10. Implement Virtualization strategy related to resources.

Project Based Learning - Provisional List of Projects

1. Explore the architectures, features, and functions of open-source operating systems
2. Design the Processes and thread management with deadlock's, synchronization
3. Design Pre-emptive Priority Scheduling algorithm implementation in any language.
4. Java program to analyse page fault for a given page frame using NRU with paging.
5. The project on simulating the multiprogramming of a specific operating system and dealing with CPU scheduling and Job scheduling.
6. Design the project that computes FCFS, SSTF, and SCAN disk-scheduling algorithms
7. Operating Systems mini project to explore the different algorithms of main memory page replacement
8. Develop a client server application to show the inter process communication.
9. Build a file system using the FUSE library.
10. Write a shell interpreter for LINUX.

Syllabus for Unit Tests:

Unit Test -1	Unit – I, Unit – II, Unit - III
Unit Test -2	Unit – IV, Unit – V, Unit - VI

Database Management System

Teaching Scheme		Examination Scheme		Credit Scheme	
	Hours/Week		Marks		Credits
Lecture:	03 Hours/Week	University Examination:	60 Marks		
Practical:	02 Hours/Week	Internal Assessment:	40 Marks	Lecture	03
		Term Work & Practical	50 Marks	Practical	01
		Total	150 Marks	Total	04

Course Objective:
Introduction to databases mainly focus on relational models and relation database design. The course enables students with the knowledge models, design paradigms and structured query language. This course introduces students to Semantic Modelling, principles of database management systems (DBMS), DBMS architecture, Database Design, data storage and query processing and transaction management. Further, the course also introduces advanced database systems.

Prerequisite:
Mathematics for Computing-I, Data Structure and Algorithmic Thinking.

Course Outcomes: On completion of the course, students will have the ability to:

1. Design a Relational Database by applying the principles of Database Design
2. Compare and Contrast File Processing and Database Processing
3. Convert the Database Design into Relational Tables as per the application requirement
4. Apply the normalisation to Database Design.
5. Use and relate the concept of transaction, concurrency control and recovery in database.
6. Write queries and commands using Structured query language (SQL)

Unit I Overview of Database Systems: **06 Hours**
Knowing Database and Database Management system (DBMS), Role and Advantages of DBMS, Problems with Traditional File System Processing, types of databases, Database System Architecture - Conceptual Level, External Level, Internal Level, Mappings, Database Users, 2 Tier Architecture and 3 Tier Architecture. Introduction to Relational Databases: Relations and Relvars, Defining Relations, Optimisation.

Unit II Relational Model: **06 Hours**
Values Vs Variables, Types Vs Representations, Type Definition, Operators, Type Generators, Tuples, relation types, Relation Values, Relation variables, Relational Algebra - Syntax and Semantics, Operations, Relational Calculus - Tuple Calculus, Calculus Vs Algebra, Integrity - Predicates and Propositions, Checking Constraints, Constraint classification scheme, Views - Definition, View retrievals and Updates

Unit III Database Design: **06 Hours**
Semantic Modelling - Entity Relationship and Extended Entity Relationship model, Functional Dependencies - Trivial and non-trivial dependencies, Closure of a set of dependencies, closure of a set of attributes, Boyce Codd Normal Form, Normalisation - 1NF, 2NF, 3NF, BCNF, higher Normal Forms.

Unit IV Storage and Querying: **06 Hours**
JBOD, RAID, Files, Data Dictionary storage, Storage Access, Indexing & Hashing - Basics, Ordered Indices, B+ Tree index Files, B Tree Index Files, Multiple Key Access, Static Hashing and Dynamic Hashing, Bitmap Indices.

Querying - Measures of Query Cost, Selection Operation, Sorting, Join Operation.

Unit V Transaction Management: **06 Hours**

Transactions Overview, Transaction Properties, Transaction Log, Concurrency control - Lost Updates, Uncommitted Data, the scheduler, Locking Methods, Time Stamping Methods, Recovery, Isolation Levels, System Recovery, Media Recovery, Savepoints, Serializability.

Unit VI Overview of Advanced Databases: **06 Hours**

Object Oriented Database, Distributed Databases, Logic Based Databases, Temporal Databases, Decision support systems. The Information System Design: System Development Life Cycle, Database Lifecycle, Conceptual Design, Logical Design, Database Design Strategies, Centralized vs Decentralised Design, NoSQL

Textbooks

1. Database System Concepts by Abraham Silberschatz, Henry F. Korth, and S. Sudarshan, 6th Edition, McGraw-Hill Education, 2010.

Reference Books

1. C.J. Date, An Introduction to Database Systems, 8/e, Pearson Education, 2004.
2. Peter Rob and Carlos Coronel, Database System- Design, Implementation and Management (7/e), Cengage Learning, 2007.

List of Assignments

The assignments will be based on a mini project for developing a Database for a system like Student Database Management system, Online Retail Database, Medical record Database etc. There will be tasks that will be allocated to the students like

1. Use the Relational Algebra.
2. Design the EER model
3. Convert the EER model to Relational Tables
4. Apply the Normalisation
5. Create the Tables using SQL and using graphical database building tools.
6. Populate and retrieve the data from tables based on requirements.

List of Laboratory Exercises

1. To study about the Database Management Software Tools and Structured Query Language
2. To Write Data Definition Language queries.
3. To Write Data Query Language queries.
4. To Write Data Control Language queries.
5. To query relational tables using nested queries and Aggregate functions.
6. To perform queries using views.
7. To demonstrate the different types of Joins.
8. Find Results for the questions asked related to the given relational Schema: Employee and Department.
9. Find Results for the questions asked related to the given relational Schema: Publications.
10. To Use MongoDB and perform CRUD operations on it.

Project Based Learning - Provisional List of Projects

RDBMS Design and implementation of various Management database systems:

1. Medical Health record management system

2. Patient detail management system
3. Student Management System
4. On-Demand Online Video Streaming
5. Sports
6. Finances Management System
7. Grocery Management System
8. Weather Management System
9. Web Database system
10. E-commerce Database system.

Syllabus for Unit Tests:

Unit Test -1	Unit – I, Unit – II, Unit - III
Unit Test -2	Unit – IV, Unit – V, Unit - VI

Wireless Communication

Teaching Scheme		Examination Scheme		Credit Scheme	
	Hours/Week		Marks		Credits
Lecture:	04 Hours/Week	University Examination:	60 Marks	Lecture	04
Practical:	02 Hours/Week	Internal Assessment:	40 Marks		
		Term Work & Oral	50 Marks	Practical	01
		Total	150 Marks	Total	05

Course Objective:
The course focuses on the fundamentals of wireless communications and provides an overview of existing and emerging wireless communication Technology and networks.

Prerequisite:
Computer Network, Physics for Computing Systems

Course Outcomes: On completion of the course, students will have the ability to:

1. Use of basic concepts and principles of wireless communication
2. Illustrate knowledge of Medium Access Control
3. Recite the working of Telecommunication Systems
4. Recite the functioning and use the satellite and broadcast systems
5. Apply and use Wireless networks.
6. Illustrate the working of Mobile IP and TCP in wireless communication.

Unit I Introduction to Wireless Communication **08 Hours**

Wireless Communication Applications, Open research topics, reference model, Frequencies for Radio transmission – regulations, Signals, Antennas, Signal Propagation – Path Loss of radio signals, Signal propagation effects, multi-path propagation, multiplexing – Space Division Multiplexing, Frequency division multiplexing, time division multiplexing, code division multiplexing.
Modulation – Amplitude shift keying, Frequency shift keying, Phase shift keying, Multi carrier modulation.
Spread Spectrum – Direct sequence spread spectrum; Frequency hopping spread spectrum.

Unit II Medium Access Control **08 Hours**

Specialized MAC requirement, Hidden terminal, and exposed terminals, near and far terminals, SDMA, FDMA, TDMA – Fixed TDM, Classical ALOHA, Slotted ALOHA, Carrier Sense multiple access, Demand assigned multiple access, packet reservation multiple access, Reservation TDMA, Multiple access with collision avoidance, Polling, CDMA – Spread Aloha multiple access, Comparison of SDMA/TDMA/FDMA/CDMA

Unit III Telecommunication Systems **08 Hours**

GSM – Mobile services, System architecture, Radio Interface, Protocols, Localisation & calling, Handover, Security.
DECT – System architecture, Protocol architecture
UMTS and IMT – 2000 – UMTS system architecture, UMTS radio interface

Unit IV Satellite Systems and Broadcasting Systems **08 Hours**

Primer – Orbital aspects, GEO, LEO, MEO, Line of Sight, Routing, Localisation, Handover, Examples.
Broadcast Systems – Cyclical repletion of data, Digital Audio broadcasting – Multimedia object transfer protocol, Digital Video broadcasting – data broadcasting, high speed internet access.

Unit V Wireless LAN **08 Hours**

Infra-red Vs Radio Transmission, Infrastructure and Ad-hoc network, IEEE 802.11 – System Architecture, Protocol Architecture, Physical layer, Medium Access control layer, MAC management, 802.11a, 802.11b.
Bluetooth – Architecture, Radio Layer, Baseband Layer, Link Manager protocol, L2CAP, Security, IEEE 802.15.

Unit VI Mobile Communication Layers **08 Hours**

Mobile IP – Entities and terminologies, IP packet delivery, Agent discovery, Registration, Tunnelling, and encapsulation, IPV6, DHCP.
Traditional TCP – Congestion control, slow start, fast retransmit/fast recovery, Indirect TCP, Snooping TCP, Mobile TCP.
Introduction to 4G, LTE network and 5G communication.

Textbooks

1. Mobile Communications, 2nd Edition, Jochen H. Schiller, Pearson Education, 2003

Reference Books

1. Wireless Communication, Theodore S. Rappaport, Prentice Hall
2. Andreas.F. Molisch, ―Wireless Communications‖, John Wiley – India, 2006.
3. Wireless Communications and Networking, Vijay Garg, Elsevier
4. Wireless Communication –Andrea Goldsmith, Cambridge University Press, 2011
5. David Tse and Pramod Viswanath, ―Fundamentals of Wireless Communication, Cambridge University Press, 2005
6. Dennis Roddy, Satellite Communications Systems, John Wiley & Sons, Ltd 5th Edition

List of Assignments

1. Discuss Comparison of Wireless Technologies in Industrial Application
2. What medium access methods are in use today in wireless networks.
3. Discuss any one System architecture of any latest telecommunication System.
4. Discuss any one latest Digital Audio broadcasting techniques
5. Elaborate latest Protocol Architecture used in practical world.
6. Introduction to 5G networks in Mobile Communication.

List of Laboratory Exercises

1. Introduction to wireless simulation.
2. Configuring wireless networking devices using simulation tool.
3. Establishing wireless LAN network using simulation tool.
4. Configuring wireless router using simulation tool.
5. Case studies on different wireless generations
6. Case studies on IEEE 802.11 Wireless LAN
7. Case studies on different wireless generations.
8. Case studies on short range wireless network
9. Realistic Studies on Wireless Structural Control.
10. Case Studies of Wireless LAN Problems.
11. Case studies on IEEE 802.11 Wireless LAN
12. Case Study of Security Issue in the Wireless Communication System.

Project Based System - Provisional List of Projects

1. Vehicle Tracking System.
2. Accident Identification System.
3. Wireless Camera Position System.
4. Remote Home Security System.

5. Wireless Voting Machine.
6. Wireless Security System.
7. Video Signal Transmitter.
8. Audio Signal Transmitter.
9. RFID based Ambulance Flashing Light with Beeper
10. Bluetooth based Garage Door Opening

Syllabus for Unit Tests:

Unit Test -1	Unit – I, Unit – II, Unit - III
Unit Test -2	Unit – IV, Unit – V, Unit - VI

Design Thinking and Communication

Teaching Scheme		Examination Scheme		Credit Scheme	
	Hours/Week		**Marks**		**Credits**
Lecture:	**02 Hours/Week**	**Term Work & Practical**	**100 Marks**	**Lecture**	**02**
Practical:	**04 Hours/Week**			**Practical**	**02**
		Total	**100 Marks**	**Total**	**04**

Course Objective:
This course focuses on enabling students to use design thinking as a problem-solving tool that includes principles like Human centred, Collaborative teamwork, learning by doing, embrace experimentation, understand patterns relationship and Systems, visualise and present.

Prerequisite:
-

Course Outcomes: On completion of the course, students will have the ability to:

1. Use the design thinking approach.
2. Understand and represent the unique needs of customers
3. Apply Empathy and visual thinking within the context of design thinking.
4. Create physical prototypes / a visual representation of an idea
5. Apply Design thinking for problem solving
6. Create a design thinking culture to drive innovation in an organisational setup

Unit I Design Thinking Background **04 Hours**

Definition of Design Thinking, Business uses of Design Thinking, Variety within the Design Thinking Discipline, Design Thinking Mindset.
Fundamental Concepts within the context of design thinking: Empathy, Ethnography, Divergent Thinking, Convergent thinking, Visual Thinking, Assumption testing, Prototyping, Validate

Unit II Resources and Processes **04 Hours**

Design Thinking Resources: Human resources, Preferred space, Materials commonly used, dynamic between design thinking teams & organisation.
Design Thinking Processes: Variety of design thinking approaches – Double Diamond approach, d.school 5-stage approach, designing for growth approach, role of Project management within design thinking.

Unit III Process Stages of Designing for Growth **04 Hours**

Practice "What Is", "What If", "What wows", "What works" process stage activities to develop a product.
Customer requirement - Knowing Your Users Identifying Needs of Customers Designing with Empathy Designing for Diversity & Inclusion, Customer needs and market, Types of product users Customer needs analysis

Unit IV Design Thinking Tools and Methods **04 Hours**

Purposeful Use of Tools and Alignment with Process – Visualization, Journey Mapping, Value Chain Analysis, Mind Mapping, Brainstorming, Concept Development, Assumption Testing, Rapid Prototyping, Customer Co-creation, Learning Launch.

Unit V Design for Services **04 Hours**

New Product Development Processes, Design for Products & Services, Sustainability through Design Thinking, Lean & Agile Product/ Service Design, Service development process, Service cycle experience map, Product vs. service systems, Service innovation examples.

Unit VI Innovation **04 Hours**

Product Development Processes Innovation Project Lifecycle, Innovation Management Models, Frugal Innovation, Entrepreneurship Vs Intrapreneurship, Innovation for Growth Data-driven Innovation
Systematic innovation process: Altitude case study.

Textbooks

1. Designing for growth: A design thinking tool kit for managers", by Jeanne Liedtka and Tim Ogilvie., 2011, ISBN 978-0-231-15838-1
2. The design thinking playbook: Mindful digital transformation of teams, products, services, businesses and ecosystems", by Michael Lewrick, Patrick Link, Larry Leifer., 2018, ISBN 978-1-119-46747-2
3. Presumptive design: Design provocations for innovation", by Leo Frishberg and Charles Lambdin., 2016, ISBN: 978-0-12-803086-8

Reference Books

1. "Systems thinking: Managing chaos and complexity: A platform for designing business architecture.", "Chapter Seven: Design Thinking", by Jamshid Gharajedaghi, 2011, ISBN 978-0-12-385915-0
2. "Cross-Cultural and User-Centred Design Thinking in a Global Organization: A Collaborative Case Analysis.", by Abildgaard, Sille Julie J., and Bo T. Christensen., She Ji: The Journal of Design, Economics, and Innovation 3.4 (2018): 277-289.
3. "Design Thinking: A Method or a Gateway into Design Cognition?.", by Gabriela Goldschmidt, She Ji: The Journal of Design, Economics, and Innovation 3.2 (2017): 107- 112

List of Laboratory Exercises

1. The Laboratory Exercises will consist of case studies related to problem solving through design thinking. Some of the Case studies are given below:
2. Design thinking applied in product creation like, Mobile and TVs
3. Problem statement like the following will be given to students. For the given case study and simple problems, the students should be able to:
    a. Complete Design Research plan
    b. Identifying insights and opportunities
    c. Perform Ideation and prototype
    d. Test Final ideas
    e. Peer feedback and group discussion
    f. Presentation
    g. Journey mapping and idea generation
    h. Develop Presentations in small groups.
    i. Final Presentations
4. Which principles and approaches in DT can be found in more holistic human-centred software development approaches and how do they differ?
5. How can artefacts with similar purposes, but different forms, be integrated?
6. How can problems be efficiently classified?
7. What are typical project situations which influence the choice of a strategy?

8. Which methods in DT can be found in / reused for other software engineering disciplines (e.g., HCI, TDD)? How do these methods differ? How can they be integrated?
9. How can artefacts, roles, and methods be seamlessly integrated? Which artefacts do overlap? Are shifts in roles and responsibilities necessary? How can milestones be efficiently defined?
10. How can resulting processes be integrated (into the overall life cycle) - for instance SCRUM? How can resulting processes be tailored?
11. The pandemic is forcing us re-evaluate safety of public health vis-a-vis the urban design, public spaces, mobility, products, and services etc. Their existing design does not ensure safety against the risk of virus transmission. Even as the threat of pandemic looms large, scenes of uncontrolled public gathering is common.
12. How to enhance the everyday commuter experience at the Mumbai Local network of Indian Railways (one of the largest and oldest in the country)

Project Based Learning - Provisional List of Projects

1. Demonstrates with a real time scenario, how understanding your customer, or subject, can completely change the perception towards a problem.
2. Show how design thinking can help redefine the values.
3. Using a real time scenario, show how problems affecting diverse groups of people.
4. Show how Issues relating to college culture can be fixed through design thinking.
5. How design thinking helps in educational advances. Cite some real time scenarios.
6. How design thinking enables Entrepreneurial initiatives. Cite some real time scenarios.
7. How design thinking is used for bringing about innovation. Illustrate with real time scenario.
8. Research how design thinking is helping medical breakthroughs. Illustrate with real time scenario.
9. How design thinking reduces the risk associated with launching new ideas, products, and services. Illustrate with real time scenario.
10. Research How to use Design Thinking for software development projects.

Programming Technologies and Tools Laboratory - IV

Teaching Scheme		Examination Scheme		Credit Scheme	
	Hours/Week		Marks		Credits
Practical:	04 Hours/Week	Term Work & Practical	50 Marks	Practical	02
		Total	50 Marks	Total	02

Course Objective
The course enables students to implement object-oriented designs with Java.

Prerequisite
Programming Technologies and Tools Laboratory – I, Programming Technologies, and Tools Laboratory – II, Paradigms of Programming.

Course Outcomes: On completion of the course, students will have the ability to:

1. Analyse the basics of Java programming.
2. Identify class, objects for real time problems.
3. Make use of constructor, Garbage Collector, and methods of string class.
4. Explore the concept of inheritance and polymorphism with the help of real time applications.
5. Handle the exception with exception handling mechanism and multi-threading programming.
6. Design the graphical user interface by using Applets and AWT.

Unit I JAVA Primer **08 Hours**

Introduction: History and Features of Java, Internals of Java Program, Difference between JDK, JRE and JVM, Internal Details of JVM.
Basics of Java Language-Variable and Reserve / Keywords present in Java, Primitive Data types, Java Operators, Decision making and branching statements in Java

Unit II Classes, Objects and Methods: **08 Hours**

Creating a Class, Visibility/Access Modifiers, Encapsulation, Methods: Adding a Method to Class, returning a Value, adding a Method That Takes Parameters, 'this' Keyword, Method Overloading, Object Creation, Using Object as a Parameters, Returning Object, Array of Objects, Memory Allocation: 'new', Static Data Members, Static Methods,

Unit III Constructors, Destructors and String Handling **08 Hours**

Use of Constructor, Characteristics of Constructors, Types of Constructors, Constructor Overloading, Constructor with Default Arguments, Symbolic Constants, Garbage Collection, Destructors and Finalizers.
String Handling: String: Immutable String, String Comparison, String Concatenation, Substring, Methods of String class, String Buffer class, StringBuilder class, Creating Immutable class, to String method

Unit IV Inheritance and Polymorphism **08 Hours**

Use of Inheritance, Types of Inheritance in Java, Role of Constructors in inheritance, Polymorphism in OOP, Types of Polymorphism, static and dynamic polymorphism, Overriding Super Class Methods. Use of "super" keyword. Interfaces, Implementing interfaces.

Unit V Exception Handling and Multithreaded programming **08 Hours**

Exception Handling: try and catch block, catch block, Nested try, finally block, throw keyword, Exception Propagation, throws keyword, Exception Handling with Method Overriding, Custom Exception.

Introduction to threads, life cycle of a thread, thread states, thread properties, methods in Threads and Runnable, setting priority of threads, synchronization and inter thread communication Life Cycle of a Thread

Unit VI Designing Graphical User Interfaces in Java **08 Hours**

Applet and its use Design Patterns using Applet and JApplet. Run Applet application by browser and applet tool. Applet Architecture. Parameters to Applet Life Cycle of Components and Containers Basics of Components Using Containers Layout Managers and userdefined layout. BorderLayout, FlowLayout, GridLayout, GridbagLayout, BoxLayout. AWT Components Adding a Menu to Window Extending GUI Features Using SWING Components Designing GUI. Advanced swing components like JProgressbar, JSlider, JRadioButton , JTree, JTable, JToggleButton.

Textbooks

1. E. Balaguruswamy, "Object Oriented Programming Using C++ and Java", Tata McGrawHill
2. Steven Holzner et al. "Java 2 Programming", Black Book, Dreamtech Press, 2009.

Reference Books

1. Java The complete reference, Herbert Schildt, McGraw Hill Education (India) Pvt. Ltd. 9th edition, 2014, ISBN: 978-0-07-180856-9 (E-book)
2. Object-Oriented Design Using Java, Dale Skrien, McGraw-Hill Publishing, 2008, ISBN - 0077423097, 9780077423094
3. Mitsunori Ogihara, "Fundamentals of Java Programming", Springer; 2018, ISBN 978-3-319-89490-4
4. Brahma Dathan Sarnath Ramnath, "Object-Oriented Analysis, Design and Implementation an Integrated Approach", Springer; 2nd ed. 2015, ISSN 1863-7310 ISSN 2197-1781 (electronic) Undergraduate Topics in Computer Science ISBN 978-3-319-24278-1, ISBN 978-3-319-24280-4.
5. T. Budd (2009), An Introduction to Object Oriented Programming, 3rd edition, Pearson Education, India.
6. J. Nino, F. A. Hosch (2002), An Introduction to programming and OO design using Java, John Wiley & sons, New Jersey
7. Y. Daniel Liang (2010), Introduction to Java programming, 7th edition, Pearson education, India

List of Laboratory Exercises

1. Write a Java program that prompts the user for an integer and then prints out all prime numbers up to that integer.
2. Write a Java program that describes a class person. It should have instance variables to record name, age, and salary. Create a person object. Set and display its instance variables.
3. Write a Java program that creates a class circle with instance variables for the centre and the radius. Initialize and display its variables
4. Write a Java program that counts the number of objects created by using static variable.
5. Write a Java program to demonstrate the constructors in java.
6. Write a Java program to demonstrate the constructor overloading.
7. Write a Java program to display the use of this keyword

8. Write a Java program that checks whether a given string is a palindrome or not.
9. Write an application that creates an interface′ and implements it
10. Write a program that can count the number of instances created for the class.
11. Write an application that executes two threads. One thread displays —Every 1000 milliseconds and other displays —B every 3000 milliseconds. Create the threads by extending the Thread class.
12. Create an abstract class shape. Let rectangle and triangle inherit this shape class. Add necessary functions.
13. Write an application that shows the usage of try, catch, throws and finally.
14. Write an Applet that displays —Hello World‖ (Background colour-black, text colour-blue and your name in the status window.)
15. Develop a scientific calculator using swings.

Project Based Learning - Provisional List of Projects

1. Airline reservation system
2. Course management system
3. Data visualization software
4. Electricity billing system
5. e-Healthcare management system
6. Email client software
7. Library management system
8. Network packet sniffer
9. Online bank management system
10. Online medical management system