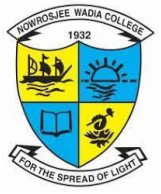

Modern Education Society's

Nowrosjee Wadia College

(AUTONOMOUS)

NAAC Accredited A+ with CGPA 3.51

Affiliated to the
Savitribai Phule Pune University
(Formerly University of Pune)

Four Year B.Sc. (Honors) Degree Program in Computer Science
(Formerly known as B. C.S.)
(Faculty of Science & Technology)
B.Sc. (Computer Science)

National Education Policy (NEP) Syllabus
(Amended in third BOS Meeting)
To be implemented from Academic Year 2024-2025

Title of the Course: B.Sc. (Computer Science)

Preamble for the Syllabus:

In compliance with the directives from the University Grants Commission, under the autonomous status of the college, the syllabus for four-year B. Sc. Computer Science at the undergraduate level is revised and reframed as per the National Educational Policy (NEP 2020) curriculum framework. Nowrosjee Wadia College has decided to change the syllabi for the degree of B.Sc. from June 2023, as the college has already shifted to the autonomous status from the academic year 2022-2023. The present syllabus is prepared by the Board of Studies in Computer Science, Nowrosjee Wadia College, taking into consideration the present relevance and application of the various branches of Computer Science. While preparing this syllabus the U.G.C. model curriculum (LOCF) and existing syllabus given by Savitribai Phule Pune University is followed.

It aims to provide students with thorough knowledge of theoretical and practical aspects of Computer Science. The objective of the course is to prepare students to undertake careers involving problem solving using Computer Science and technologies, or to pursue advanced studies and research in Computer Science. The syllabus which comprises of Computer Science subject along with that of the three allied subjects (Mathematics, Electronics and Statistics) covers the foundational aspects of computing sciences and also develops the requisite professional skills and problem solving abilities using Computing Sciences. The program creates opportunities of hands-on learning through projects and gives knowledge and practical experience of the latest technologies. It also encourages a student to work effectively as team member and demonstrate professional behaviour. On completion of this course, a student will not only develop a diverse set of skills to prepare for higher studies in Computer Science and for employment, but will also encourage students to launch their own start-ups or venture into new types of careers using their interdisciplinary training.

Objectives:

- To develop problem solving abilities using a computer.
- To build the necessary skill set and analytical abilities for developing computer-based solutions for real life problems.
- To train students in professional skills related to the software industry.
- To prepare necessary knowledge base for research and development in Computer Science.
- To help students build a successful career in Computer Science and to produce entrepreneurs who can innovate and develop software products.

PROGRAM OUTCOMES (POs):

The Bachelor of Science with Computer Science (BSc with CS) program enables students to attain by the time of graduation following 10 PO's. The course syllabi and the overall curriculum have been designed to achieve these outcomes:

Program Outcome (PO)	Short title	Description A Graduate student in Computer Science will be able to:
PO1	Knowledge outcome	Fundamental understanding of the principles of Computer Science and its connections with other disciplines
PO2	Problem Analysis and solution	Procedural knowledge that creates different types of professionals related to Computer Science,
PO3	Development of various allied skills	Demonstrate the aptitude of Computer Programming and Computer based problem solving skills with use of basic knowledge in applied subjects Statistics, Electronics and Mathematics. Ability to link knowledge of Computer Science with other two chosen auxiliary disciplines of study.
PO4	Modern Tool usage	Display the knowledge of appropriate theory, practices and tools for the specification, design, and implementation. Ability to formulate, to model, to design solutions, procedure and to use software tools to solve real world problems and evaluate .
PO5	Environment and Sustainability	Understand, critically analyse and attempt at finding the solutions to various environmental issues and obligate themselves towards sustainable development at the local, national and global context.
PO6	Communication and Leadership	Ability to present result using different presentation tools. Communicate proficiently and develop the quality of

		presentation, good communication, leadership, working in group/team as a member, and other extracurricular activities.
PO7	Research skills and Aptitude	Understand the concept of research, general research methods and is able to analyse, interpret and draw rational inferences. Ability to pursue higher studies of specialization. Ability to appreciate emerging technologies and tools.
PO8	Ethics	Understand professional ethics and human values. Display ethical code of conduct in usage of Internet and Cyber systems.
PO9	Societal Applications	Ability to formulate, to model, to design solutions, procedure and to use software tools to solve real world problems and evaluate. Attempts at analysing and inspecting varied socio-economic issues in computer culture perspective by applying the knowledge to the societal issues.
PO10	Life Skills	Recognise the scope of computers in terms of exploring the career opportunities, employment and life-long engagement in teaching and utilise the knowledge for publication for future academic endeavours.

Eligibility for the Course

XIIth Science or its equivalent examination with Mathematics. All other criteria are same as per Savitribai Phule Pune University rules.

Teaching Scheme:

- The course is a 3 year, 6 semesters full time under graduate course and 1 more year, 2 semester B. Sc. (CS) Honors course.
- The course follows the NEP pattern as per Savitribai Phule Pune University.
- 1 credit theory = 15 hours
- 1 credit practical = 30 hours = 4 hrs week per batch

MODERN EDUCATION SOCIETY'S

NOWROSJEE WADIA COLLEGE, PUNE

(An Autonomous College Affiliated to Savitribai Phule Pune University)

NEP Course Structure and other details

For

B. Sc.(COMPUTER SCIENCE)

(Based on NEP 2020 framework)

(To be implemented from the Academic Year 2024-25)

Structure showing Credits in each semester for a four year Under Graduate programme B. Sc. (Computer Science)

SEM	MAJOR		MINOR	OE	VSC/SEC		OJT/FP/CEP/CC/RP	AEC/VEC/IKS	TOTAL
	Mandatory	Elective			VSC	SEC			
I	6 (2T+1P)	0	0	4	2 ST:P	2 ST:T	2 CC 2	6 AEC 2, VEC 2, IKS 2	22
II	6 (2T+1P)	0	2 (EL:T)	4 = (2 + 2)	2 ST:P	2 M:T	2 CC 2	4 AEC 2, VEC 2	22
III	8 (3T+1P)	0	4 (EL: T + L)/ (MT: T + L)	2	2	0	4 FP 2, CC 2	2 AEC	22
IV	8 (3T+1P)	0	4 (EL: T + L)/ (MT: T + L)	2	0	2	4 CEP 2 CC 2	2 AEC	22
V	10 (3T+2P)	4CS(1T+1P)	4 S/M/E	0	2 CS:P	0	2 FP/CEP	0	22
VI	10 (3T+2P)	4CS(1T+1P)	2 S/M/E	0	2 CS:P	0	4 OJT	0	22
TOTAL	48	8	16	12	10	6	18	14	132
VII	12 (2T+2P)	4 CS(1T+1P)	4 RM	0	2	0	0	0	22
VIII	12 (2T+2P)	4 CS(1T+1P)	0	0	2	0	4 OJT	0	22

Abbreviations used throughout -

OE : Open Elective , VSC : Vocational Skill Courses, VEC: Value Education Courses, CC : Co-Curricular Courses, AEC: Ability Enhancement Course, IKS : Indian Knowledge System, OJT : On Job Training, FP : Field Project, CEP : Community Engagement Project
ST : Statistics, EL : Electronics, M : Mathema

EVALUATION PATTERN :

Note : The Department follows all rules, regulations and procedure related to the examination decided by Examination Section of college.

(i) Each course shall be evaluated with Continuous Evaluation (CE) and Semester-end Examination (SEE) mechanism.

(ii) Theory courses: Continuous Evaluation shall be of 15 marks and Final Assessment shall be of 35 marks.

(iii) Practical courses: Continuous Evaluation shall be of 15 marks and Final Assessment shall be of 35 marks.

(iv) To pass a course of 2 credits, a student has to earn minimum 20 marks out of 50, provided that he/she should earn minimum 6 marks in Continuous Evaluation (out of 15) and minimum 14 marks (out of 35) in End-Semester Examination. That is passing criterion is minimum 40% marks in the examination.

(v) For Continuous Evaluation (out of 15 marks), There has to be one written test of 10 marks (Mid-Semester Examination). The remaining 05 marks shall be based on the continuous evaluation consisting of tutorial, viva, seminars, home-assignments, mini project, survey, group discussion etc.(on approval of Head of the Department) and performance and attendance in the lectures and labs.

(vi) There shall be revaluation of the answer scripts of End-Semester Examination (out of 35 marks) of theory papers only, but not of internal assessment papers as per Ordinance No. 134 A and B.

PATTERN OF THE QUESTION PAPER

Continuous Evaluation (Theory papers)

(1) As a part of Internal Evaluation, there shall be written test (Mid-Semester Examination) of 10 marks. Pattern of the question paper is as follows.

(2) Continuous Evaluation (Mid-Semester Examination of 10 marks, Duration: 45 minutes)

Question No.	Total Marks	No. Of questions	Remarks
Q. 1.	05	Solve any 5 out of 7	Short answer / objective type of questions. Each question

			carry 1 mark.
Q. 2.	05	Solve any 1 out of 2 questions	Each question carry 5 marks

(3) Semester-End Examination for B. Sc. courses, out of 35 marks, shall be of 2 hours duration. The pattern of the question paper will be as decided by the examination section of the college.

AWARD OF GRADES AND GRADE POINTS

The mapping of percentage to letter grade and grade point (for each course) is given in the following Table.

Sr. No.	Grade Letter	Grade Point	Marks
1.	O (Outstanding)	10	45 ≤ Marks ≤ 50
2.	A+ (Excellent	9	40 ≤ Marks ≤ 44
3.	A (Very Good)	8	35 ≤ Marks ≤ 39
4.	B+ (Good)	7	27.5 ≤ Marks ≤ 34
5.	B (Above Average)	6	25 ≤ Marks < 27.5
6.	C (Average)	5	22.5 ≤ Marks ≤ 24
7.	D (Pass)	4	20 ≤ Marks < 22.5
8.	F (Fail)	0	Marks < 20
9.	Ab (Absent)	0	

CGPA :The CGPA is the weighted average of the grade points obtained in all courses (theory and Practicals) by a student in all the courses in 6 semesters.

Based on the performance of the student in the Semester Examinations, Nowrosjee Wadia College will declare the results and issue the Semester Grade sheets. Also, the College will declare the results and issue the Grade sheets at the end of the course. The class will be awarded to a student on the basis of CGPA. The award of the class shall be as per the following table

Sr. No.	CGPA	Class of the degree awarded
1	9.50 or more than 9.50	OUTSTANDING (O)
2	8.50 or more but less than 9.50	EXCELLENT (A+)

3	7.50 or more but less than 8.50	VERY GOOD (A)
4	6.25 or more but less than 7.50	GOOD (B+)
5	5.25 or more but less than 6.25	ABOVE AVERAGE (B)
6	4.75 or more but less than 5.25	AVERAGE (C)
7	4.00 or more but less than 4.75	PASS (D)

Semester 1 (First Year)

Course Type	Course	Course / Paper Title	Hours / Week	Credit	CIA	ESE	Total
Major Mandatory (4 + 2)	Major Paper 1 (Theory)	Computational Thinking	2	2	15	35	50
	Major Paper 2 (Theory)	Fundamentals of Databases	2	2	15	35	50
	Major Paper 3 (Practical)	Laboratory Course on Computational Thinking and Fundamentals of Databases	4	2	15	35	50
Major Electives							
Minor	NIL	NIL	NIL	NIL	NIL	NIL	NIL
OE (2 +2)		Open Elective 1 Open Elective 2	2 2	2 2	15 15	35 35	50 50
VSC (2)	Major Specific Practical 1	Laboratory Course on Basic Statistics for Computer Science	2	2	15	35	50
SEC (2)	Skill Paper 1 (Theory)	Basic Statistics for Computer Science	2	2	15	35	50
AEC(2),	English Theory	English Communication I	2	2	15	35	50
VEC (2)	EVS Theory	Environment Science I	2	2	15	35	50
IKS (2)	Major Specific Theory	Vedic Mathematics and Computer Fundamentals	2	2	15	35	50
CC (2)	CC-I Course	Physical Education / Cultural Activities, NSS/NCC and Fine/ Applied/ Visual/ Performing Arts Course	2	2	15	35	50

Semester 2 (First Year)

Course Type	Course	Course / Paper Title	Hours /Week	Credit	CIA	ESE	Total
Major Mandatory (4 + 2)	Major Paper 4 (Theory)	Advanced 'C' Programming	2	2	15	35	50
	Major Paper 5 (Theory)	Advanced Database Management Systems	2	2	15	35	50
	Major Paper 6 (Practical)	Laboratory Course on Advanced 'C' Programming and Advanced Database Management Systems	4	2	15	35	50
Major Electives							
Minor	Minor Paper 1 (Theory/Practical)	Fundamentals of Electronics (T/P)	2	2	15	35	50
OE (2 +2)		Open Elective 3 Open Elective 4	2 2	2 2	15 15	35 35	50 50
VSC (2)	Major Specific Practical II	Advanced Statistics for Computer Science	2	2	15	35	50
SEC (2)	Skill Paper 2 (Theory)	Basic Mathematics for Computer Science	2	2	15	35	50
AEC(2),	English Theory	English Communication II	2	2	15	35	50
VEC (2)	EVS Theory	Environment Science II	2	2	15	35	50
IKS (2)							
CC (2)	CC-II Course	Physical Education / Cultural Activities, NSS/NCC and Fine/ Applied/Visual/ Performing Arts Course	2	2	15	35	50

Semester 3 (Second Year)

Course Type	Course	Course / Paper Title	Hours /Week	Credit	CIA	ESE	Total
Major Mandatory (6+2)	Major Paper 7 (Theory)	Data Structures and Algorithms-I	2	2	15	35	50
	Major Paper 8 (Theory)	Software Engineering	2	2	15	35	50
	Major Paper 9 (Theory)	Computer Networks	2	2	15	35	50
	Major Paper 10 (Practical)	Laboratory Course on Data Structures and Algorithms and Software Engineering	4	2	15	35	50
Major Electives							
Minor (4)	Minor Paper 2 (Theory) Minor Paper3 (Practical)	Digital Electronics and Circuit Design (T) / Mathematics (T) Laboratory Course on Sensors and IoT / Laboratory Course in Mathematics	2 4	2 2	15 15	35 35	50 50
OE (2)		Open Elective 5	2	2	15	35	50
VSC (2)	Major Specific Theory	Basics of Web Designing	2	2	15	35	50
SEC (2)							
AEC(2),	MIL	MIL-I (Hindi) / MIL-I (Marathi)	2	2	15	35	50
VEC (2)							
IKS (2)							
FP/CEP (2)	FP –I	Field Project Related to Computer Science	6	2	15	35	50
CC(2)	CC III	Physical Education / Cultural Activities, NSS/NCC and Fine/ Applied/ Visual/ Performing Arts Course	2	2	15	35	50

Semester 4 (Second Year)

Course Type	Course	Course / Paper Title	Hours /Week	Credit	CIA	ESE	Total
Major Mandatory (6+ 2)	Major Paper 11 (Theory)	Data Structures and Algorithms-II	2	2	15	35	50
	Major Paper 12 (Theory)	Principles of Multimedia	2	2	15	35	50
	Major Paper 13 (Theory)	Network Security	2	2	15	35	50
	Major Paper 14 (Practical)	Data Structures and Algorithms-II And Computer Networks	4	2	15	35	50
Major Electives							
Minor (4)	Minor Paper 4 (Theory) Minor Paper 5 (Practical)	Digital Electronics and Circuit Design(T) / Mathematics (T) Laboratory Course on Sensors and IoT / Laboratory Course in Mathematics	2 4	2 2	15 15	35 35	50 50
OE (2)		Open Elective 6	2	2	15	35	50
VSC (2)							
SEC (2)	Skill Paper 3 (Theory)	Python Programming	2	2	15	35	50
AEC(2),	MIL	MIL-I (Hindi) / MIL-I (Marathi)	2	2	15	35	50
VEC (2)							
IKS (2)							
CEP(2)	CEP –I		6	2	15	35	50
CC(2)	CC-IV	Physical Education / Cultural Activities, NSS/NCC and Fine/ Applied/ Visual/ Performing Arts Course	2	2	15	35	50

Semester 5 (Third Year)

Course Type	Course	Course / Paper Title	Hours /Week	Credit	CIA	ESE	Total
Major Mandatory (6+ 2+2)	Major Paper 15 (Theory)	Operating Systems -I	2	2	15	35	50
	Major Paper 16 (Theory)	Foundations of Data Science	2	2	15	35	50
	Major Paper 17 (Theory)	Object Oriented Programming using Java	2	2	15	35	50
	Major Paper 18 (Practical)	Laboratory Course on Java and Foundations of Data Science	4	2	15	35	50
	Major Paper 19 (Practical)	Laboratory Course on OS-I	4	2	15	35	50
Major Electives	Elective 1(Theory)	Theory of Computation	2	2	15	35	50
	Elective 1(Practical)	Computer Graphics using Python	4	2	15	35	50
	OR						
	Elective 2(Theory)	Web Technologies	2	2	15	35	50
	Elective 2 (Practical)	Lab course on Web Technologies	4	2	15	35	50
Minor (4)	Minor Paper 6 (Theory)	E/M/S	2	2	15	35	50
	Minor Paper 7 (Practical)	E/M/S	4	2	15	35	50
OE (2)							
VSC (2)	Major Specific Practical III	Laboratory Course on Block Chain Technology	4	2	15	35	50
SEC (2)							
AEC(2),							
VEC (2)							
IKS (2)							
FP / CEP(2)	FP –II/CEP II		6	2	15	35	50

Semester 6 (Third Year)

Course Type	Course	Course / Paper Title	Hours / Week	Credit	CIA	ESE	Total
Major Mandatory (6 +2+2)	Major Paper 20 (Theory)	Operating Systems-II	2	2	15	35	50
	Major Paper 21 (Theory)	Data Analytics	2	2	15	35	50
	Major Paper 22 (Theory)	Advanced Java Programming	2	2	15	35	50
	Major Paper 23 (Practical)	Operating Systems-II	4	2	15	35	50
	Major Paper 24 (Practical)	Laboratory Course on Advanced Java Programming	4	2	15	35	50
Major Electives	Elective 3 (Theory)	Compiler Construction	2	4	15	35	50
	Elective 3 (Practical)	Laboratory Course on Git and Github	4		15	35	50
	OR						
	Elective 4 (Theory)	Fundamentals Of Artificial Intelligence	2		15	35	50
	Elective 4 (Practical)	Laboratory Course on Software Testing Tools	4		15	35	50
Minor (2)	Minor Paper 8 (Theory)	E/M/S	2	2	15	35	50
VSC (2)	Major Specific Practical IV	Laboratory Course on Linux Administration	4	2	15	35	50
SEC (2)							
AEC(2),							
VEC (2)							
IKS (2)							
OJT	FP –II/CEP II		12	4	15	35	50

Semester 7 (Fourth Year)

Course Type	Course	Course / Paper Title	Hours / Week	Credit
Major Mandatory (4 + 4 +2+ 2)	Major Paper 25 (Theory)	Design and Analysis of Algorithms	4	4
	Major Paper 26 (Theory)	Advanced Operating Systems	4	4
	Major Paper 27 (Practical)	Principles of Programming Languages	4	2
	Major Paper 28 (Practical)	Laboratory Course on Advanced Operating Systems and Principles of Programming Languages	4	2
Major Electives	Elective 5 (Theory)	Digital Forensics and Cloud Security	2	4
	Elective 5 Practical)	Laboratory Course on Digital Forensics and Cloud Security	4	
	OR			
	Elective 6 (Theory)	Mean-stack web development	2	
	Elective 6 (Practical)	Laboratory Course on Mean-stack web development lab	4	
Minor (4)	Minor Paper (Theory)	Research Methodology	4	4
OE (2)				
VSC (2)				
SEC (2)	Major Specific Practical V	CS	2	2
AEC(2),				
VEC (2)				
IKS (2)				
FP / CEP(2)				
OJT(4)				

Semester 8 (Fourth Year)

Course Type	Course	Course / Paper Title	Hours / Week	Credit
Major Mandatory (4 + 4 +2+2)	Major Paper 29 (Theory)	Web frameworks	4	4
	Major Paper 30 (Theory)	Machine Learning and Artificial Intelligence	4	4
	Major Paper 31 (Practical)	Laboratory Course on Web frameworks	4	2
	Major Paper 32 (Practical)	Laboratory Course on Machine Learning and Artificial Intelligence	4	2
Major Electives	Elective 7 (Theory)	Software Project Management	2	4
	Elective 7 (Practical)	Laboratory Course on Project Management tools	4	
	OR			
	Elective 8 (Theory)	NLP and Large Language Models	2	
	Elective 8 (Practical)	Laboratory Course on NLP and Large Language Models lab	4	
OE (2)				
VSC (2)				
SEC (2)	Major Specific Practical VI	CS(P)	4	2
AEC(2),				
VEC (2)				
IKS (2)				
FP / CEP(2)				
OJT(4)			12	4

MODERN EDUCATION SOCIETY'S

NOWROSJEE WADIA COLLEGE, PUNE

(An Autonomous College Affiliated to Savitribai Phule Pune University)

NEP Syllabus for

Second Year B.Sc. (COMPUTER SCIENCE)

(Based on NEP 2020 framework)

(To be implemented from the Academic Year 2024-25)

Course Code:
Title : Data Structures and Algorithms – I
Semester III Major paper 7 Theory

Teaching Scheme	No. of Credits	Examination Scheme
30 Hours	2	CE : 15 marks ESE: 35 marks

Prerequisites:

- Basic knowledge of algorithms and problem solving
- Knowledge of C Programming Language

Learning Objectives

1. To learn the systematic way of solving problem
2. To understand the different methods of organizing large amount of data
3. To efficiently implement the different data structures
4. To efficiently implement solutions for specific problems
5. To apply linear data structures.

Course Contents

Unit 1	Introduction to Data Structures and Algorithm Analysis	8

1.1 Introduction
- 1.1.1 Need of Data Structure
- 1.1.2 Definitions - Data and information, Data type, Data object, ADT, Data Structure
- 1.1.3 Types of Data Structures

1.2 Algorithm analysis

1.3 Space and time complexity, Graphical understanding of the relation between different functions of n, examples of linear loop, logarithmic, quadratic loop etc.

1.4 Best, Worst, Average case analysis, Asymptotic notations (Big O, Omega Ω, Theta □), Problems on time complexity calculation.

Unit 2	Array as a Data Structure	10

2.1 ADT of array, Operations

2.2 Array applications – Searching, Sequential search, Binary Search

2.3 Sorting Terminology- Internal, External, Stable, In-place Sorting

2.3.1 Comparison Based Sorting - Lower bound on comparison based sorting, Methods- Bubble Sort, Insertion Sort, Selection Sort, Algorithm design strategies - Divide and Conquer strategy, Merge Sort, Quick Sort, Non Comparison Based Sorting: Counting Sort, Radix Sort.

Unit 3	**Linked List**	**6**
3.1 List as a Data Structure, differences with array. 3.2 Types of Linked List – Singly, Doubly, Circular 3.3 Operations on Linked List - create, traverse, insert, delete. 3.4 Applications of Linked List – polynomial representation, Addition of two polynomials		
Unit 4	**Stack**	**6**
4.1 Introduction 4.2 Operations – init(), push(), pop(), isEmpty(), isFull(), peek(), time complexity of operations. 4.3 Implementation- Static and Dynamic with comparison 4.4 Applications of stack 4.4.1 Function call and recursion, String reversal, palindrome checking 4.4.2 Expression types - infix, prefix and postfix, expression conversion and evaluation (implementation of infix to postfix, evaluation of postfix) 4.4.3Backtracking strategy - 4 queens problem (implementation using stack)		

Learning Outcomes

On completion of the course, student will be able to

1. To use well-organized data structures in solving various problems.
2. To differentiate the usage of various structures in problem solution.
3. Implementing algorithms to solve problems using appropriate data structures.

Learning Resources

1. Classic Data Structures, D. Samanta, Prentice Hall India Pvt. Ltd, 2nd Edition.
2. Fundamentals of Data Structures in C, Ellis Horowitz, Sartaj Sahni,Susan Anderson-Freed,Universities Press, 2nd Edition.
3. Data Structures using C and C++, YedidyahLangsam, Moshe J. Augenstein, Aaron M.Tenenbaum, 2nd Edition, Pearson Education.
4. Data Structures: A Pseudo code approach with C, Richard Gilberg ,Behrouz A. Forouzan,Cengage Learning, 2nd Edition.

Course Code: Title : Software Engineering Semester III (Major Paper 8) Theory		
Teaching Scheme 30 Hours	**No. of Credits** 2	**Examination Scheme** CE : 15 marks ESE:35marks

Prerequisites

- Problem solving using computer.

Learning Objectives

1. To get knowledge and understanding of software engineering discipline.
2. Know the role played by requirements analysis in requirement integration.
3. Appreciate the idea behind Design Patterns in handling common problems faced during Building an application.
4. To learn analysis and design principles for software project development

Course Contents

Unit1	**Introduction to Software Engineering and Process Models**	**4**

1.1 Definition of Software
1.2 Nature of Software Engineering
1.3 Changing nature of software
1.4 Software Process
 1.4.1 The Process Framework
 1.4.2 Umbrella Activities
 1.4.3 Process Adaptation
1.5 Generic Process Model
1.6 Prescriptive Process Models
 1.6.1 The Waterfall Model
 1.6.2 Incremental Process Models
 1.6.3 Evolutionary Process Models

Unit2	**Agile Development**	**4**

2.1 What is Agility?
2.2 Agile Process
2.2.1 Agility Principles
2.2.2 The Politics of Agile Development
2.2.3 Human Factors
2.3 Scrum
2.4 Other Agile Frameworks
2.4.1 Extreme Programming(XP)- XP Values, XP Process, Industrial XP
2.4.2 Kanban
2.4.3 DevOps

Unit3	Requirements Analysis	3

3.1 Requirement Elicitation,
3.2 Software requirement specification (SRS)
3.2.1 Developing Use Cases (UML)
3.3 Building the Analysis Model
3.3.1 Elements of the Analysis Model
3.3.2 Analysis Patterns
3.3.3 Agile Requirements Engineering
3.4 Negotiating Requirements
3.5 Validating Requirements

Unit4	Requirements Modeling	13

4.1 Introduction to UML
4.2 Structural Modeling
4.2.1 Use case model
4.2.2 Class model
4.3 Behavioral Modeling
4.3.1 Sequence model
4.3.2 Activity model
4.3.3 Communication or Collaboration model
4.4 Architectural Modeling
4.4.1 Component model
4.4.2 Artifact model
4.4.3 Deployment model
4.5 Data Modeling Concepts- Data Objects, Data Attributes and Relationship
4.6 Design Process
4.6.1 Software Quality Guidelines and Attributes
4.6.2 Evolution of Software Design
4.7 Design Concepts
Abstraction, Architecture, Patterns, Separation of Concerns, Modularity, Information Hiding, Functional Independence, Refinement, Aspects, Refactoring, Object Oriented Design Concepts, Design Classes

Unit 5	Design for Mobility	6

5.1 The Challenges
- 5.1.1 Development Considerations
- 5.1.2 Technical Considerations

5.2 Mobile Development Life Cycle
- 5.2.1 User Interface Design

5.3 Mobile Architectures
5.4 Context Aware Apps
5.5 Web Design Pyramid
5.6 WebApp Interface Design
- 5.6.1 Aesthetic Design
- 5.6.2 Content Design
- 5.6.3 Architecture Design
- 5.6.4 Navigation Design

Learning Outcomes

On completion of this course, students will be able to:

1. Compare and chose a process model for a software project development.
2. Identify requirements analyze and prepare models.
3. Prepare the SRS, Design document, Project plan of a given software system

Learning Resources

1.Software Engineering: A Practioner's Approach, Roger S. Pressman, 9th Edition, Tata McGraw-Hill

2. Software Engineering Concepts, Richard Fairley, Tata McGraw-Hill.

3.Information Technology Project Management, Kathy Schwalbe, 6th Edition,Thomson Publication

4.Software Engineering, Ian Sommerville,v ,10th Edition Pearson publication

Subject Code: **Subject Title: Computer Networks** **Semester III (Major Paper 9) Theory**		
Teaching Scheme 30 Hours	**No. of Credits** 2	**Examination Scheme** CE : 15 marks ESE:35marks

Learning Objectives

1. To understand basic terms of computer networks and the internet environment.
2. To prepare students with basic networking concepts: data communication, protocols and standards, various topologies and applications of network.

Course Contents

Unit 1	**Introduction to Networks and Network Models**	**6**

1.1 Data communication, components, data representation
1.2 Networks, network criteria, network types - LAN, WAN, Switching, The Internet,Accessing the Internet
1.3 Network Software- Protocol hierarchies, Design Issues of the layer, Connection-Oriented and Connectionless Services
1.4 Reference models - OSI Reference Models, TCP/IP Reference model, Connectiondevices in different layers, Comparison of OSI and TCP/IP Reference Models.
1.5 History and evolution of internet.

Unit 2	**Physical and Data Link Layer**	**8**

2.1 Communication at the physical layer, data rate limits - Noiseless channel (Nyquist bitrate), noisy channel (Shannon capacity), Performance - bandwidth, throughput,Latency, bandwidth-delay product, jitter.
2.2 Design issues of Data Link Layer, Services - Framing, flow control, error control, congestion control, Link layer addressing Framing Methods – Character Count, Flag bytes with Byte Stuffing, Flags bits with Bit Stuffing, Physical Layer Coding Violations
2.3 The Channel allocation problem, Static and dynamic allocation, Media Access Methods - Taxonomy of multiple-access protocols
2.4 Switching and TCP/IP layers, Types - circuit switching, packet switching andmessage switching
2.5 Wired LANs - Standard Ethernet characteristics, Addressing, Access method,implementation, Fast and Gigabit Ethernet
2.6 Wireless LANs - Architectural comparison, Characteristics, Access control, IEEE 802.11 architecture, Bluetooth architecture

Unit 3	**Network Layer**	**8**

3.1 Network layer services - Packetizing, Routing and forwarding, other services
3.2 Open and closed loop congestion control.
3.3 IPv4 addressing- Address space, classful addressing, Subnetting, Supernetting, Classless addressing, Network address resolution (NAT).
3.5 Forwarding of IP packets- based on destination address, based on label.
3.6 Network Layer Protocols- Internet Protocol (IP), IPv4 datagram format, Fragmentation,options
3.7 Mobile IP-addressing, agents, Three phases
3.8Next Generation IP- IPv6 address representation, address space, address types, IPv6 protocol, packet format, extension header, Difference between IPv4 and IPv6
3.9Routing - General idea, Algorithms - Distance vector routing, link state routing, path vector routing

Unit 4	**Transport Layer**	**8**

4.1 Transport layer Services- Process-to-process communication, Addressing, Encapsulation and decapsulation, Multiplexing and demultiplexing, Flowcontrol, Pushing or pulling, Flow control, Buffers, Sequence numbers, Acknowledgements, sliding window, congestion control.
4.2 Connectionless and Connection-oriented service, Port numbers
4.3 Transport layer protocols- User datagram protocol, user datagram, UDPServices
4.4 Transmission Control Protocol - TCP Services, TCP Features, TCP Segment format, three-way handshake for connection establishment and termination, State transition diagram, windows in TCP.

Learning Outcomes

On completion of this course, students will be able to :
1. Have a good understanding of the OSI and TCP/IP Reference Models.
2. Understand the working of various layers and protocols in the layers.
3. Analyze the requirements for a given organizational structure and select the most appropriate networking architecture and technologies.

Learning Resources

1. Computer Networks, Andrew S. Tanenbaum, 5th Edition, Pearson Education
2. Data Communication and Networking, Behrouz Fourouzan, 5th Edition, McGraw HillPvt. Ltd.

Course Code: **Title : Practical course on Data Structures and Algorithms I & Software Engineering** **Semester III Major paper 10**		
Teaching Scheme 4 Hours/per week/per batch Batch Size : 12	**No. of Credits** 2	**Examination Scheme** CE : 15 marks ESE: 35 marks

Operating Environment:
For Data Structures:

- **Operating system:** Linux
- **Editor:** Any linux based editor like vi, gedit etc.
- **Compiler** : cc or gcc

Lab Book:
The lab book is to be used as a hands-on resource, reference and record of assignment submission and completion by the student. The lab book contains the set of assignments which the student must complete as a part of this course.

Programming Assignments:
Programs should be done individually by the student intheir respective login. The codes should be uploaded on either the local server, Moodle, Github or any open source LMS. Print-outs of the programs and output may be taken but not mandatory for assessment.

Assessment:
Continuous assessment of laboratory work is to be done based on overall performance and lab assignments performance of student. Each lab assignment assessment will be assigned grade/marks based on parameters with appropriate weightage. Suggested parameters for overall assessment as well as each lab assignment assessment include-timely completion, performance, innovation, efficient codes and good programming practices.

- **Internal Evaluation :**
  - 10 marks will be given based on a mini project of Software Engineering.
  - 5 marks will be allocated for Assignment completion and practical attendance.
- **University Evaluation :**
  - The Practical slip will be of 35 Marks which will be based on Data structures.

Suggested Assignments for Data Structures – I

Assignment1: Searching Algorithms

Implementation of searching algorithms to search an element using: Linear Search, Binary Search (with time complexity)

Assignment 2:Sorting Algorithms - I

Implementation of sorting algorithms: Bubble Sort, Insertion Sort

Assignment 3:Sorting Algorithms - II

Implementation of sorting algorithms: Quick Sort, Merge Sort

Assignment 4:Singly Linked List

1.Dynamic implementation of Singly Linked List to perform following operations: Create, Insert, Delete, Display, Search

Assignment 5:Doubly Linked List

1.Dynamic implementation of Doubly circular Linked List to perform following operations: Create, Insert, Delete, Display, Search

Assignment 6:Linked List Applications

1. Addition of two polynomials in a single variable.

Assignment 7:Stack

1. Static and Dynamic implementation of Stack to perform following operations: Init, Push, Pop, Peek, Isempty, Isfull

Assignment 8:Applications of Stack

1. Implementation of an algorithm that reverses string of characters using stack and checks whether a string is a palindrome.
2. Infix to Postfix conversion.

Evaluation of postfix expression.

Suggested Assignments for Software Engineering mini Project	**3**

1. Prepare detailed statement of problem for the selected mini project
2. Identify suitable process model for the same.
3. Develop Software Requirement Specification for the project.
4. Identify scenarios and develop UML Use case
5. Other artifacts: Class Diagram, activity diagram, sequence diagram, component diagram and any other diagrams as applicable to the project.

Sample project titles: (These are just samples, students are suggested to take up different case studies)

1. Online mobile recharge system
2. Credit calculation system
3. Image sharing and editing system
4. Internal examination system
5. e-learning management system

Subject Code:
Subject Title: Digital Electronics and Circuit Design
Semester III/IV Minor Paper 3 and Paper 5 Electronics Theory

Teaching Scheme	**No. of Credits**	**Examination Scheme**
30 Hours	2	CE: 15 marks ESE: 35 marks

Learning Objectives

1. Understand digital circuit designing concept
2. Study basics of Computer organization and Architecture
3. Know the principles of digital communication
4. Features of 8086 and 8051

Course Contents

Unit 1	**Digital circuit design**	**10**

1.1 Introduction to Boolean algebra and rules of Boolean algebra
1.2 Simplification of expressions using rules of Boolean algebra, Concept of SOP and POS, Maxterm and Minterm
1.3 Introduction to K maps: Concept of Cells, Pairs, Quads, Octets
1.4 Combinational circuit
1.4.1 Half adder and Full adder
1.4.2 Designing 4 bit Binary to Gray and Gray to Binary convertor
1.5 Sequential circuit Design
1.5.1 Definition and Truth tables of JK flipflop, D flipflop anFlipflop
1.5.2 Excitation table for JK flipflop, and T flipflop
1.5.3 Designing of 3 bit synchronous Up counter
1.5.4 Designing of 3 bit Synchronous Down counter

Unit 2	**Introduction to Computer Organization**	**6**

2.1 Block diagram and function of each block of Computer system
2.2 Block diagram of a CPU and function of its blocks
2.3 Concept of Buses and registers
2.4 Stack: Need and its organization
2.5 I/O interfacing concepts
2.5.1Block diagram and need of I/O interface
2.5.2 Programmed I/O part, DMA (Definition, Types of DMA transfer and DMA controller)

Unit 3	**Introduction to Microprocessors and Microcontrollers**	**7**
3.1 Definition of Microprocessors and Microcontrollers 3.2 Differences between Microprocessors and Microcontrollers 3.3 Microprocessor architectures: Von Neumann and Harvard architecture (in brief) 3.4 Block diagram of 8086 microprocessor and function of each block 3.5 Concept of Superscalar architecture and pipelines 3.6 Block diagram of 8051 Microcontroller and function of each block 3.7 Concept of Parallel processing and Multicore processors		
Unit 4	**Digital Communication**	**7**
4.1 Block diagram of Communication system and function of each block 4.2 Concept of Channel capacity, bandwidth, bit rate, baud rate and signal to noise ratio 4.3 Sampling theorem: Nyquist and Shannon's theorem (Statement and formula) 4.4 Modulation and Demodulation 4.4.1 Concept of modulation and demodulation 4.4.2 Digital modulation techniques: PCM, Concept of ASK,FSK and PSK techniques 4.5 Concept of FDM, TDM and WDM		
Learning Outcomes On the completion of the course student will be able to: 1. Understand the basics of digital circuit design. 2. Design digital circuits. 3. Differentiate between microprocessor and microcontroller. 4. Understand the requirements of digital communication.		
Learning Resources 1. 8051 microcontroller and embedded system using assembly and C: Mazidi and McKinley,2nd Edition, Pearson publications 2. Modern Digital Electronics by R.P Jain 5th edition McGraw Hill publications 3. Digital fundamentals by Floyd 11th edition Pearson publications 4. Microprocessor and interfacing, Douglas Hall 3rd edition Mc Graw Hill publication 5. Communication Electronics Frenzel 3rd edition, McGraw Hill 6. Digital communications Kumar M Satish PHI learning Pvt Ltd.		

Subject Code:
Subject Title: Laboratory Course on Digital Electronics and Circuit Design
Semester III/IV Minor Paper 4 and Paper 6

Teaching Scheme	**No. of Credits**	**Examination Scheme**
4 Hours/per week/per batch	2	CE: 15 marks ESE: 35 marks

Learning Objectives

1. Illustrate various communication circuits using CRO.
2. To design, build and test circuits used in digital communication.
3. Understand the difference between sequential and combinational circuits.

Course Contents

Any eight from the following

1. Study of 3 bit Pulse Code Modulation technique
2. Study of Frequency Shift Keying
3. Study of Time Division Multiplexing
4. Study of Frequency Division Multiplexing
5. Study of Error detection and correction using Hamming Code technique
6. PN Sequence Generator
7. Study of PAM
8. Design of Half and Full adder circuit using K map
9. Design of 4 bit binary to gray and Gray to binary convertor using K map
10. Design of 3 bit Synchronous Up/Down/Updown counter using K map
11. Study of Flipflops
12. Study of Shift register(SISO and Ring counter)
13. Study of diode matrix ROM
14. Study of 4 to 1 Multiplexer and 1 to 4 Demultiplexer
15. Study of read write action of RAM 7489

Learning Outcomes

On the completion of the course student will be able to:

1. Design and build common combinational and sequential digital circuits.
2. Understand digital communication concepts by using circuits
3. Learn to use measuring instruments like CRO, Signal generator and Multimeter

Subject Code: Subject Title: Applied Mathematics Semester III/IV Minor Paper 3 and Paper 5 (Theory)		
Teaching Scheme 30 Hours	**No. of Credits** 2	**Examination Scheme** CE: 15 marks ESE: 35 marks

Learning Objectives

1. Introduction of fundamental concepts and theory of computer graphics.
2. 2D transformation curves and introduction to 3D transformation.
3. Find optimal solution to problems.
4. Various method that are used for quantitative decision making

Course Contents

Unit 1	**Two dimensional transformations**	**8**

1.1 Introduction.
1.2 Representation of points.
1.3 Transformations and matrices.
1.4 Transformation of points.
1.5 Transformation of straight lines
1.6 Midpoint Transformation
1.7 Transformation of parallel lines
1.8 Transformation of intersecting lines
1.9 Transformation: rotations, reflections, scaling, shearing.
1.10 Combined transformations.
1.11 Transformation of a unit square.
1.12 Solid body transformations.
1.13 Translations and homogeneous coordinates.
1.14 Rotation about an arbitrary point.
1.15 Reflection through an arbitrary line.

Unit 2	**Three dimensional Transformations**	**6**

2.1 Introduction.
2.2 Three dimensional – Scaling, shearing, rotation, reflection, translation.
2.3 Multiple transformations.
2.4 Rotation about – an axis parallel to coordinate axes, an arbitrary axis in space
2.5 Reflection through an Arbitrary plane

Unit 3	**Linear Programming Problem I**	8

3.1 Development of Operations Research.
3.2 Definition of Operations Research

3.3 Characteristics of operations Research
3.4 Introduction of linear Programming
3.5 Requirements of a linear programming problem
3.6 Areas of applications of linear Programming
3.7 Formulation of linear programming problems.
3.8 Graphical Method of a solution
3.9 Some Exceptional cases
3.10 The General linear Programming Problem
3.11 Canonical and Standards Forms of linear Programming Problem
3.12 Theory of Simplex Method

Unit 4	**Transportation Models**	8

4.1 Introduction to the model
4.2 Assumptions in the Transportation Model
4.3 Definition of the transportation Problems.
4.4 Matrix Terminology
4.5 Formulation and Solution Transportation Models
4.6 Variants in Transportation Problems
4.7 Additional problems

Learning Outcomes

On the completion of the course student will be able to:

1. Evaluate 3D transformations.
2. Demonstrate knowledge of key notions and principles related to 2 dimensional transformations.
3. Identify the role of Linear programming problem solving skill in real life business model.
4. Develop mathematical and computational modeling of real decision making problems.

Learning Resources

1. Mathematical elements for Computer graphics, F. Rogers, J. A. Adams, Mc Graw Hill 2nd Edition.
   Unit 1: Chapter 2: Sec. 2-1 to 2.17,
   Unit 2: Chapter 3: Sec. 3.1 to 3.1
2. Operations Research,.Prem Kumar Gupta ,D.S. Hira , Revised edition.
   Unit 3 :Chapter 1: Sec 1.1 to 1.3 ,Chapter 2 : Sec 2.1, 2.2, 2.5, 2.6, 2.9 to 2.13
   Unit 4: Chapter 3: Sec 3.1 to 3.7
3. Computer Graphics with OpenGL, Donald Hearn, M. Pauline Baker, Warren Carithers,Pearson, 4th Edition
4. Operations Research and Introduction by H. A. Taha. Pearson Publication, 9th Edition.
5. Operations Research by Panneerselvam, Prentice Hall of India, 2nd Edition.
6. Operations Research, Theory and Applications by J. K. Sharma, Trinity publication, 6th Edition

Subject Code: **Subject Title: Mathematics Practical Python Programming Language-I** **Semester III/IV Minor Paper 4 and Paper 6 (Practical)**		
Teaching Scheme 4 Hours/ per week/per batch	**No. of Credits** 2	**Examination Scheme** CE: 15 marks ESE: 35 marks

Learning Objectives

1. The programming language Python and its application in mathematical computations.
2. The basic syntax and features of Python programming, as well as mathematical concepts like linear algebra and numerical methods.

Course Contents

Unit 1	**Introduction to Python**	

1.1 Installation of Python
1.2 Values and types: int, float and str,
1.3 Variables: assignment statements, printing variable values, types of variables.
1.4 Operators, operands and precedence:+, -, /, *, **, % PEMDAS(Rules of precedence)
1.5 String operations: + : Concatenation, * : Repetition
1.6 Boolean operator:
1.7 Comparison operators: ==, !=, >, =, <=
1.8 Logical operators: and, or, not
1.9 Mathematical functions from math, cmath modules.Keyboard input: input() statement Division Algorithm (without Proof)

Unit 2	**String, list, tuple**	

2.1 Strings
- 2.1.1 Length (Len function)
- 2.1.2 String traversal: Using while statement, Using for statement
- 2.1.3String slice
- 2.1.4Comparison operators (>, Binary Operation)

2.2 Lists
- 2.2.1 List operations
- 2.2.2 Use of range function
- 2.2.3 Accessing list elements
- 2.2.4 List membership and for loop
- 2.2.5 List operations
- 2.2.6 Updating list: addition, removal or updating of elements of a list

2.3 Tuples
- 2.3.1 Defining a tuple,

2.3.2 Index operator,
2.3.3 Slice operator,
2.3.4 Tuple assignment,Tuple as a return value

Unit 3	**Iterations and Conditional statements**	

3.1 Conditional and alternative statements, Chained and Nested Conditionals: if, if-else, if-elif-else, nested if, nested if-else
3.2 Looping statements such as while, for etc, Tables using while.
3.3 Functions
3.3.1 Calling functions: type, id
3.3.2 Type conversion: int, float, str
3.3.3 Composition of functions
3.3.4 User defined functions, Parameters and arguments

Unit 4	**Linear Algebra**	

4.1 Matrix construct, eye(n), zeros(n, m) matrices
4.2 Addition, Subtraction, Multiplication of matrices, powers and invers of a matrix.
4.3 Accessing Rows and Columns, Deleting and Inserting Rows and Columns
4.4 Determinant, reduced row echelon form, null space, columnspace, Rank
4.5 Solving systems of linear equations (Gauss Elimination Method, Gauss Jordan Method, LU-decomposition Method)
4.6 Eigenvalues, Eigen vectors.

Unit 5	**Numerical methods in Python**	

5.1 Roots of Equations
5.2 Newton-Raphson Method
5.3 False Position (Regula Falsi) Mehtod
5.4Numerical Integration
5.4.1 Trapezoidal Rule,
5.4.2 Simpson's 1/3rd Rule,
5.4.3 Simpson's 3/8th Rule

Practical's :

1 Introduction to Python, Python Data Types-I (Unit 1)
2 Python Data Types- II (Unit 2)
3 Control statements in Python-I (Unit 3- 3.1, 3.2)
4 Control statements in Python-II (Unit 3- 3.3)
5 Application: Matrices (Unit 4 – 4.1-4.3)
6 Application : Determinants, system of Linear Equations (Unit 4- 4.4, 4.5)
7 Application : System of equations (Unit 4- 4.5)
8 Application : Eigen values, Eigenvectors (Unit 4 – 4.6)

9 Application : Roots of equations (Unit 5 – 5.1)
10 Application : Numerical integration (Unit 5 – 5.2, 5.3)
11 Application : Numerical integration (Unit 5 – 5.4)

Learning Outcomes

Upon completing this course, students should be able to:

1. Understand the basic syntax and programming concepts of Python.
2. Use Python to perform basic mathematical computations, such as addition, subtraction,multiplication, and division.
3. Use Python to manipulate strings, lists, and tuples.
4. Understand and use conditional statements and loops in Python.
5. Understand and use functions in Python.
6. Use Python to perform linear algebra operations such as matrix addition, subtraction, multiplication, and inversion.
7. Understand and apply numerical methods in Python for finding roots of equations and for numerical integration
8. Solve systems of linear equations using various methods such as Gauss Elimination GaussJordan and LU-decomposition.

Learning Resources

1 How to think like a Computer Scientist: Learning with Python, Downey, A. et al., John Wiley, 2015.
2 Introduction to Scientific Computing in Python, Robert Johansson, Apress, 2nd Edition.
3 Fundamentals of Python - First Programs, Lambert K. A., Cengage Learning India,2015.
4 Introduction to Computing and Programming in Python, Guzdial, M. J., Pearson, India, 2012.
5 Introduction to Computing Using Python, Perkovic, L., John Wiley, 2nd edition, 2015.
6 Python Programming: An Introduction to Computer Science, Zelle, J., Franklin,Beedle & Associates Inc., 2004.
7 Matplotlib for Python Developers, Sandro Tosi,Packet Publishing Ltd.(2009)

Course Code: Title : Basics of Web Designing Semester III (VSC-2)		
Teaching Scheme 30 Hours	**No. of Credits** 2	**Examination Scheme** CE : 15 marks ESE: 35 marks

Prerequisites

- Basic knowledge of the internet.

Learning Objectives

1. To explain different components and technologies of web applications.
2. To develop simple web pages using HTML, CSS.
3. To build dynamic web pages with validation using Java Script.

Course Contents

Unit 1	**Web Design Principles**	**4**

1.1 Introduction to Internet and World Wide Web
1.2 Client Server Architecture
1.3 Components of Web Application: Front End, Back End, Communication Technologies
1.4 Web Server Vs Web browser
1.5 Website: working, types
1.6 Domain Name Server and Uniform Resource locator
1.7 Basic principles involved in developing a web site
1.8 Five Golden rules of web designing
1.9 Web Standards and W3C recommendations

Unit 2	**Introduction to HTML5**	**10**

2.1 HTML and its features.
2.2 What is difference between HTML and HTML5
2.3 Basic structure of an HTML document
2.4 Block level Tags and Inline Tags
2.5 Core Attributes of HTML- id, style, class, title
2.6 Working with Lists, Tables, Frames and Iframes
2.7 Working with Text and Image Hyperlinks
2.8 Working with Forms and Controls
2.9 HTML multimedia.
2.10 Responsive web design

Unit 3	**Introduction to Cascading Style Sheets**	**10**

3.1 Concept of CSS

3.2 Need of CSS
3.3 Creating Style Sheet -Inline, External, Embedded CSS.
3.4 CSS Selectors
3.5 CSS Layouts
3.6 CSS Styling Properties (Background, Text Format, Controlling Fonts)
3.7 CSS Box Model, navigation bar

Unit 4	**JScript**	**6**

4.1 Overview of JavaScript
4.2 JavaScript Basic Syntax(JS datatypes, JS variables)
Primitives, Operations and Expressions , Screen Output and keyboard input(Verification and Validation)
4.3 JS Control statements and JS Functions
4.4 JavaScript HTML DOM Events(onmouseup, onmousedown, onclick, onload, onmouseover, onmouseout).
4.5 JS Strings and JS String methods
4.6 JS popup boxes(alert, confirm, prompt).

Hands on Exercises:

1. Creating HTML Pages to implement Block Level Tags.
2. Creating HTML Pages to implement Inline Level Tags.
3. HTML Programming using list and Table tag.
4. HTML Programming using Image and Hyperlink tag.
5. HTML Programming using Frame, IFrame and Form Tag
6. HTML programs using Inline, Internal and External style sheets.
7. Program using HTML and CSS to implement CSS Properties like – CSS Colors, CSS Border, CSS Padding, CSS position
8. Program to create a CSS document on placing Images at different positions.
9. Programs using HTML and CSS to implement CSS Layout Property.
10 .Java Script program using conditional statements and loops.
11. Java Script program demonstrating window object: ‘alert box’,‘prompt box’and ‘Confirm box’.
12. Java Script program demonstrating use of HTML events like ‘OnClick’ ,’Onkeydown’ etc .
13. Java Script program demonstrating Geo Location.

Learning Outcomes : On completion of this course, students will be able to :

1. Understand the fundamentals of the Internet and how the web functions.
2. Design a static webpage by applying HTML elements.
3. Apply CSS concepts for designing HTML web pages.
4. To create dynamic web pages by using HTML,CSS and JavaScript.
5. To acquire knowledge and skills for creation of web site considering client-side programming.

Learning Resources
1.HTML5 and CSS3- WAN BAYROSS - BPB Publications 2.HTML and CSS –Thomas A Powell, 5th Edition 3. Head First HTML and CSS - Elisabeth Robson and Eric Freeman, 2nd Edition, O'Reilly 4. An introduction to Web Design + Programming - Paul S Wang, Sanda S. Katila 5. Java Script -A beginners guide - John Pollock, 3rd Edition, McGraw Hill Web References 1. http://www.w3schools.com 2. https://html5andcss3.org/

SEMESTER -IV

Course Code: **Title : DATA STRUCTURES AND ALGORITHMS-II** **Semester IV Major paper 11**		
Teaching Scheme 30 Hours	**No. of Credits** 02	**Examination Scheme** CE : 15 marks ESE: 35 marks

Prerequisites

- Knowledge of C Programming Language
- Basic knowledge of algorithms
- Basic knowledge of linear data structures

Learning Objectives

1. To learn the systematic way of solving problems
2. To design algorithms
3. To understand the different methods of organizing large amount of data
4. To efficiently implement the non-linear data structures

Course Contents

Unit 1	**Queue**	**6**

1.1 Introduction
1.2 Operations - init(), enqueue(), dequeue(), isEmpty(), isFull(), peek(),time complexity of operations, differences with stack.
1.3 Implementation - Static and Dynamic with comparison
1.4 Types of Queue - Linear Queue, Circular Queue, Priority Queue, Double Ended Queue (with implementation)
1.5 Applications – CPU Scheduling in multiprogramming environment, Round robin algorithm

Unit 2	**Tree**	**10**

2.1 Concept and Terminologies
2.2 Types of Binary trees - Binary tree, skewed tree, strictly binary tree, full binary tree, complete binary tree, expression tree, binary search tree, Heap
2.3 Representation – Static and Dynamic
2.4 Implementation and Operations on Binary Search Tree - Create, Insert, Delete, Search, Tree traversals– preorder, inorder, postorder (recursive implementation)
2.5 Balanced tree- AVL trees- Rotations
2.6 Applications of trees- Heap sort(implementation)

Unit 3	**Graph**	**6**

3.1Concept and terminologies
3.2 Graph Representation –Adjacency matrix, Adjacency list, Inverse Adjacency list, Adjacency multilist

3.3Graph Traversals – Breadth First Search and Depth First Search (with implementation) 3.4Applications of graph, Topological sorting, Single source shortest path - Dijkstra's algorithm 3.5 Use of graphs in social networks		
Unit 4	**Hash Table**	**8**
4.1 Concept of Hashing 4.2 Terminologies – Hash table, Hash function, Bucket, Hash address, collision, synonym, overflow etc. 4.3 Properties of good hash function 4.4 Hash functions : division function, MID square , folding methods 4.5 Collision resolution techniques 4.5.1 Open Addressing - Linear probing, quadratic probing, rehashing 4.5.2 Chaining		
Learning Outcomes On completion of this course students will be able to 1. Implementation of different data structures efficiently 2. Usage of well-organized data structures to handle large amount of data Usage of appropriate data structures for problem solving		
Learning Resources 1. Classic Data Structures, D. Samanta, Prentice Hall India Pvt. Ltd, 2nd Edition. 2. Fundamentals of Data Structures in C, Ellis Horowitz, Sartaj Sahni,Susan Anderson-Freed,Universities Press, 2nd Edition. 3. Data Structures using C and C++, YedidyahLangsam, Moshe J. Augenstein, Aaron M.Tenenbaum, 2nd Edition, Pearson Education. 4. Data Structures: A Pseudo code approach with C, Richard Gilberg ,Behrouz A. Forouzan,Cengage Learning, 2nd Edition.		

Subject Code: Subject Title: Principles of Multimedia Semester IV (Major Paper 12) Theory		
Teaching Scheme 30 Hours	**No. of Credits** 2	**Examination Scheme** CE: 15 marks ESE: 35 marks

Learning Objectives

1. To introduce basic components of multimedia
2. To understand different multimedia file formats
3. To identify emerging trends in multimedia
4. To understand the role of compression in multimedia
5. To identify theoretical and practical aspects in designing multimedia systems

Course Contents

Unit 1	**Introduction to Multimedia**	**5**

1.1 Goals, objectives, and characteristics of multimedia
1.2 Multimedia building blocks: text, image, audio, video, animation,
1.3 Multimedia Applications
1.4 Multimedia architecture
1.5 Evolving Technologies for Multimedia Systems
1.6 Editing, and Authoring tools
1.7 Multimedia Trends

Unit 2	**Text and Image Processing**	**10**

2.1 Text: Types of text, Text compression, Text file formats
2.2 Image: Image Data Representation, Image Acquisition, Image Processing, Image File formats, Image compression: Types of Compression: Lossless: RLE, Shannon - Fano algorithm, Arithmetic coding. Lossy: Vector quantization, Fractal Compression Technique, Transform coding and Hybrid: JPEG-DCT

Unit 3	**Audio and Video Processing**	**10**

3.1 Audio: Nature of sound waves, Digital audio, Components of audio system, Digital audio processing, file formats: AIFF, VOC, AVI, WMA, OGG, PCM,MP3,AAC Audio compression techniques: DM, ADPCM and MPEG.
3.2 Video: Digital video, Digital Video Processing, digital Video file formats: AVI, MOV,RM,WAV,FLV,3GP,Video editing, Video Compression: H-261,H-263, MPEG

Unit 4	**Animation and Virtual Reality**	**5**

4.1 Principles of Animation, Techniques of animation, Computer based Animation, Rendering Algorithms, Animation File formats, Animation tools.

4.2 Virtual Reality: Forms of Virtual Reality, VR Applications, Peripheral Devices
Learning Outcomes 1. Understand basic building blocks and applications of Multimedia. 2. Analyze different algorithms for compression 3. Apply acquired knowledge in the field of multimedia.
Learning Resources 1. Principles of Multimedia, Ranjan Parekh, 2nd Edition, TataMcGraw-Hill. 2. Multimedia Computing, Communication and Applications, Ralf Steinmetz and Klara Nahrstedt, Pearson Education. 3. Fundamentals of Multimedia, Ze-Nian Li, Marks S. Drew, Pearson Education.

Course Code: Title: Network Security Semester IV (Major Paper13) Theory		
Teaching Scheme 30 Hours	**No. of Credits** 2	**Examination Scheme** CE:15 marks ESE: 35 marks

Learning Objectives 1. Explore the different methods used for Network/INTERNET security.
2. Identify, analyze, and remediate computer security breaches.

Course Contents

Unit 1	**Application Layer**	**10**

1.1 Domain Name System
- 1.1.1 Name space-Flat name space, Hierarchical name space
- 1.1.2 Domain Name Space -Label ,Domain name, FQDN,PQDN
- 1.1.3 Distribution of Domain Name Space-Hierarchy of name servers, zone, Root server, Primary and secondary servers.
- 1.1.4 DNS in the Internet: Generic domains, Country domains,inverse domain
- 1.1.5 Resolution-Resolver,mapping names to address,mapping addresses to names, recursive resolution, iterative resolution, caching

1.2 Electronic Mail-
- 1.2.1 Architecture-First scenario, second scenario, Third scenario, Fourth scenario
- 1.2.2 User agent-services of user agent, types of UA Format of e-mail
- 1.2.3 MIME-MIME header
- 1.2.4 Message transfer agent-SMTP
- 1.2.5 Message Access Agent: POP and IMAP

1.3 File Transfer
- 1.3.1 FTP-Communication over data control connection
- 1.3.2 File type, data structure
- 1.3.3 Transmission mode
- 1.3.4 anonymous FTP

Unit 2	**Introduction to network and cyber Security**	**4**

2.1 Introduction: Network Security, Cyber Security, Differences between Cyber and Network security

2.2 Need for Security

2.3 Security Attacks
- 2.3.1Malware Attack,
- 2.3.2 Man-in-Middle Attack,
- 2.3.3Password, Attack,
- 2.3.4 Phishing

2.4 Services and Mechanisms

2.5 Network Security Model
- 2.5.1 Network Access Control (NAC)

2.5.2 Antivirus and Antimalware Software 2.5.3 Virtual Private Networks (VPN) 2.5.4 Firewall Protection, etc.		
Unit 3	**Cryptography**	**6**
3.1. Terminology: Cryptography, plain text and cipher text, cipher key, categories of 3.2. cryptography-Symmetric key, asymmetric key 3.3. Encryption model 3.4. Symmetric Key Cryptography 3.4.1 Traditional ciphers – substitution cipher, shift cipher, Transposition cipher 3.4.2 Simple Modern ciphers-XOR, Rotation cipher, s-box,p-box 3.4.3 Modern round ciphers-DES 3.4.4 Mode of operation-ECB,CBC,CFB,OFB 3.5. Asymmetric key Cryptography 3.6. Need and Principles of Public Key Cryptosystems 3.7. RSA Algorithm 3.8. Key Distribution and Management 3.9. Diffie-Hellman Key Exchange 3.10. Digital Signatures		
Unit 4	**Authentication**	**5**
4.1 Authentication Requirements 4.2 Message Authentication Codes 4.3 Hashes 4.4 MD5 & SHA 4.5 User Authentication: Password, Certificate based & Biometric Authentication 4.6 Kerberos		
Unit 5	**Security Mechanisms and Protocols**	**5**
5.1 Firewalls 5.2 IP Security 5.3 VPN 5.4 Intrusion Detection 5.5 Web Security 5.6 SSL, TLS		
Learning Outcomes On completion of this course students will be able to: 1. Develop the Concept of Security needed in the Communication of data through computers and networks along with Various Possible Attacks 2. Understand Various Encryption mechanisms for secure transmission of data and management of key required for encryption 3. Understand authentication requirements and study various authentication mechanisms 4. Understand network security concepts and study different Web security mechanisms		
Learning Resources		

1. Data communications and networking by Behrouz Forouzan 4th/5th edition McGraw Hill Pvt Ltd,
2. Computer Networks by Andrew S Tanenbaum, Pearson Education
3. Cryptography and Network Security: Principles and Practice, William Stallings,7th edition, Pearson Education
4. Network Security Essentials: Applications and Standards (For VTU), William Stallings, , 3rd edition,Pearson Education

Course Code: **Title : Practical course on Data Structures and Algorithms II & Computer Networks** **Semester IV Major paper 14**		
Teaching Scheme	**No. of Credits**	**Examination Scheme**
4 hours / per week /per batch Batch size : 12	2	CE : 15 marks ESE: 35 marks

Lab Book:

The lab book is to be used as a hands-on resource, reference and record of assignment submission and completion by the student. The lab book contains the set of assignments which the student must complete as a part of this course.

Programming Assignments:

Programs should be done individually by the student in the respective login. The codes should be uploaded on either the local server, Moodle, Github or any open source LMS. Print-outs of the programs and output may be taken but not mandatory for assessment.

Assessment:

Continuous assessment of laboratory work is to be done based on overall performance and lab assignments performance of student. Each lab assignment assessment will be assigned grade/marks based on parameters with appropriate weightage. Suggested parameters for overall assessment as well as each lab assignment assessment include-timely completion, performance, innovation, efficient codes and good programming practices.

- **Internal Evaluation :**
  - 10 marks will be given based on Networking assignments.
  - 5 marks will be allocated for Assignment completion and practical attendance
- **University Evaluation :**
  - The Practical slip will be of 35 Marks which will be based on Advanced Data structures.

Operating Environment:

For Data Structures:

- **Operating system:** Linux
- **Editor:** Any linux based editor like vi, gedit etc.
- **Compiler** : cc or gcc

Course Contents :-

Assignment 1:Linear Queue

1. Static and Dynamic implementation of linear Queue to perform following operations: Init, enqueue, dequeue Peek, IsEmpty, IsFull.

Assignment 2 :Circular and Priority Queue

1. Implementation of circular queue

Implementation of priority queue

Assignment 3: Binary Search Tree and Traversals

1. Implement Binary Search Tree (BST) to perform following operations on BST– Create, Recursive Traversals - Inorder, Preorder, Postorder
2. Perform following operations: insert, delete

Assignment 4: Applications of Binary Tree

1. Sort set of elements using Heap sort

Assignment 5: Graph implementation

1. Implement Graph as adjacency matrix and adjacency list
2. Calculate indegree and outdegree of vertices
3. Graph traversals: BFS and DFS.

Assignment 6: Networking Assignment-I

Assignment 7: Networking Assignment-II

Assignment 8: Networking Assignment-III

Course Code Course Title: Python Programming Semester IV (SEC)(Theory)		
Teaching Scheme **30 Hours**	**No. of Credits** 2	**Examination Scheme** CE: 15 marks ESE: 35 marks

Learning Objectives

1. To introduce programming concepts using python.
2. Student should be able to develop Programming logic using python.
3. To develop basic concepts and terminology of python programming.
4. To test and execute python programs.

Course Contents

Unit 1	**An Introduction to Python**	**3**

1.1 Introduction to Python-History, features, Applications, Installing Python, Running Simple Python program

1.2 Basics of Python

1.3 Standard data types - basic, none, Boolean (true & False), numbers, Variables, Constants, Python identifiers and reserved words, Lines and indentation, multi-line statements and Comments, Input/output with print and input ,functions Declaration, Operations on Data such as assignment, arithmetic, relational, logical and bitwise operations, dry run, Simple Input and output etc.

Unit 2	**Control Statements**	**5**

2.1 Sequence Control – Precedence of operators, Type conversion

2.2 Conditional Statements: if, if-else, nested if-else,

2.3 Looping- for, while, nested loops, loop control statements (break, continue, pass)

2.4 Strings: declaration, manipulation, special operations, escape character, string formatting operator, Raw String, Unicode Strings, Built-in String methods.

Unit 3	**Lists, functions, tuples and dictionaries, Sets**	**10**

3.1 Python Lists: Concept, creating and accessing elements, updating & deleting lists, traversing a List, reverse Built-in List Operators, Concatenation, Repetition, In Operator, Built-in List functions and methods.

3.2 Functions: Definitions and Uses, Function Calls, Type Conversion Functions, Math Functions, Composition, Adding New Functions, Flow of Execution, Parameters and Arguments, Variables and Parameters, Stack Diagrams, Void Functions, Anonymous, functions Importing with from, Return Values, Boolean Functions, More Recursion, Functional programming tools - filter(), map(), and reduce(),recursion, lambda forms.

3.3 Tuples and Dictionaries**:** Tuples, Accessing values in Tuples, Tuple Assignment, Tuples

as return values, Variable-length argument tuples, and Basic tuples operations, Concatenation, Repetition, in Operator, Iteration, Built-in tuple functions, indexing, slicing and matrices. Creating a Dictionary, Accessing Values in a dictionary, Updating Dictionary, Deleting Elements from Dictionary, Properties of Dictionary keys, Operations in Dictionary, Built-In Dictionary Functions, Built-in Dictionary Methods.

3.4 Sets- Definition, transaction of set(Adding, Union, intersection), working with sets

Unit 4	Modules ,Working with files, Exception handling	12

4.1 Modules: Importing module, Creating & exploring modules, Math module, Random module, Time module Packages: Importing package, creating package, examples

4.2 Working with files: Creating files and Operations on files (open, close, read, write), File object attributes, file positions, Listing Files in a Directory, Testing File Types, Removing files and directories, copying and renaming files, splitting pathnames, creating and moving directories

4.3 Regular Expression- Concept of regular expression, various types of regular expressions, using match function.

4.4 Classes and Objects: Overview of OOP (Object Oriented Programming), Class Definition, Creating Objects, Instances as Arguments, Instances as return values, Built-in Class Attributes, Inheritance, Method Overriding, Data Encapsulation, Data Hiding

4.5 Multithreaded Programming: Thread Module, creating a thread, synchronizing threads, multithreaded priority queue Modules: Importing module, Creating and exploring

4.6 Exception Handling: Built-in Exceptions, Handling Exceptions, Exception with Arguments, User-defined Exceptions.

Learning Outcomes

On completion of the course, student will be able to:

1.Develop logic for problem solving

2.Determine the methods to create and develop Python programs by utilizing the data

3. Structures like lists, dictionaries, tuples and sets.

4. To be familiar about the basic constructs of programming such as data, operations,conditions, loops, functions etc.

5.To write python programs and develop a small application project

Learning Resources

1. An Introduction to Computer Science using Python 3 by Jason Montojo, Jennifer Campbell, Paul Gries, The pragmatic bookshelf-2013

2. “Beginning Python: Using Python and Python 3.1, James Payne, Wrox Publication

3. Introduction to Problem Solving with Python by E Balguruswamy,TMH publication-2016

4. Object-oriented Programming in Python, Michael H. Goldwasser, David Letscher,Pearson Prentice Hall-2008