BHARATI VIDYAPEETH UNIVERSITY, PUNE
FACULTY OF ENGINEERING AND TECHNOLOGY

Programme: B. Tech. (I.T.) – SEM III – 2014 Course

Sr. No.	Subject	Teaching Scheme (Hrs/Week)			Examination Scheme (Marks)							Credits			
		L	P/D	T	End Semester Examination	Continuous Assessment				TW & Oral	Total	Theory	TW &Pr	TW &Or	Total
						Unit Test	Attendance	Assignments							
1	Fundamentals of Software Engineering	3	--	1	60	20	10	10	--	--	100	4	--	--	4
2	Discrete Mathematics	3	2	--	60	20	10	10	50	--	150	3	1	--	4
3	Software Project Management	3	--	--	60	20	10	10	--	--	100	3	--	--	3
4	Data Structures and Files	3	2	--	60	20	10	10	50	--	150	3	1	--	4
5	Platform Independent Programming Paradigms	3	2	--	60	20	10	10	--	50	150	3	--	1	4
6	Professional skill Development-III	4	--	--	100	--	--	--	--	--	100	4	--	--	4
7	IT Lab-I	--	4	--	-	--	--	--	50	--	50	--	2	--	2
	TOTAL	**19**	**10**	**1**	**400**	**100**	**50**	**50**	**150**	**50**	**800**	**20**	**04**	**01**	**25**

BHARATI VIDYAPEETH UNIVERSITY, PUNE

FACULTY OF ENGINEERING AND TECHNOLOGY

Programme: B. Tech. (I.T.) – SEM IV – 2014 Course

Sr.no	Subject	Examination Scheme (Marks)								Credits		
		P/D	T	End Semester Examination	Continuous Assessment	TW & Practical			TW & Oral	Theory	Tw & Or	Total
					Unit Test	Attendance	Assignments					
1.	Advanced Data Structure	2	-	60	20	10	10	--		3	1 --	4
2.	Digital Electronics and Logic Design	--	--	60	20	10	10	--		3	--	3
3.	Database Management System	2	--	60	20	10	10	50		3	-- 1	4
4.	Engineering Mathematics III	--	1	60	20	10	10	--		4	-- -	4
5.	Computer Graphics	2	--	60	20	10	10	--		3	1 --	4
6.	Professional skill Development--IV	--	--	100	-	-	-	-		4	--	4
7.	IT Lab-II	4	-	--	--	-	-	-		-	2 --	2
	TOTAL	**10**	**1**	**400**	**100**	**50**	**50**	**50**		**20**	**04 01**	**25**

Total CreditsSemester - III = 25
Total CreditsSemester - IV = 25

Grand Credits = 50

Fundamentals of Software Engineering

Teaching Scheme	Examination Scheme		Credit Allotted
Theory : 3Hrs/Week	**End Semester Examination**	**: 60 Marks**	**Theory : 3**
Tutorial : 1Hr/Week	**Continuous assessment**	**: 40 Marks**	

Course Objectives:

To enable students to work in teams and use the latest software technologies to develop and implement creative solutions to complex problems.

Course Prerequisites:

Students should have knowledge of

1) Developing well-structured, modular programs

2) Mathematical knowledge of computer science, including discrete structures, algorithms design and analysis.

Course Outcome:

Students will be able to:

1) learn appropriateness of software development model for given software system.

2) document user requirement using different communication techniques.

3) create the analysis model of the system under consideration .

4) gain knowledge of data, control and function design concepts

5) understand software quality concepts

6) Formulate a testing strategy for a software system,

UNIT-I **Introduction to Software Engineering** : **(6 Hours)**

Definition of Software Engineering, Software characteristics, Applications, Software myths, Software Development Process models: The Waterfall model, Incremental process models, Evolutionary Process models, Component based development process model, The Formal Method Model, Aspect–Oriented software Development, Unified Process, Agile Processes Models.

UNIT-II **System Engineering Practices:** **(6 Hours)**

Analysis Modeling Principles, Design Modeling Principles, Coding Principles, Testing Principles, Computer based system, System Modeling, System Simulation, System Modeling
Requirement Engineering: Requirements Engineering Tasks, Initiating Requirement engineering Process, Eliciting requirement, Introduction to SRS format, creating SRS.

UNIT-III **Analysis and modeling**: Elements of Analysis Model, Analysis modeling approaches: Data modeling, Scenario based modeling, Flow oriented modeling. **(6 Hours)**

UNIT-IV **Design Engineering:** **(6 Hours)**
System Design: Design Concepts, The Design model architecture, cohesion and coupling, Data Design, Architectural Styles and Patterns, Architechtural Design, Mapping Data flow into Software Architecture ,User Interface design.
Coding: TOP-DOWN and BOTTOM-UP structure programming, Information Hiding, Programming Style.

UNIT-V **Software Change Management :** **(6 Hours)**
Software configuration management (SCM), Elements of SCM, Base lines, Software configuration items, SCM Repository, SCM process: Version Control, Change Control, Configuration Audit, Status Reporting.

UNIT-VI **Testing Strategies:** **(6 Hours)**
Levels of Testing, Functional Testing, Structural Testing, Test Plan, Test Case Specification, Test case design, A strategic approach to software Testing: Verification and Validation Testing, Organizing for software Testing, Software Testing Strategy for conventional Architecture: Unit Testing Integration Testing, Validation Testing, System Testing, Debugging, White-box, Black-box testing, Basis path Testing, Control structure testing.

Assignements: Implement Study assignments assigned by course faculty .
Text Books:
1) Roger S. Pressman, Software Engineering: A Practitioner's Approach (6/e.) McGraw Hill, 2011.

2) James F. Peter, Software Engineering - An Engineering Approach, John Wiley (2004).

3) Pankaj Jalote, Software Engineering: A Precise Approach , Wiley India, 2010.

4) Ian Sommerville, Software Engineering, Addison-Wesley Publishing Company, (2006) 8th ed.

Reference Books:

1) *A Shalloway and J Trott, Design Patterns Explained: A new perspective on object oriented design (2/e), Pearson, 2004.*

2) *Rajib Mall, Fundamentals Of Software Engineering ,PHI Learning Pvt. Ltd 2009*

Syllabus for Unit Test:

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

Discrete Mathematics

Teaching Scheme	Examination Scheme		Credit Allotted
Theory : 3 Hrs/Week	**End Semester Examination**	**: 60 Marks**	**Theory : 3**
Practical : 2 Hrs/Week	**Continuous Assessment**	**: 40 Marks**	**Tw & Pr : 1**
	Term Work and Practical (Tw & Pr)	**: 50 Marks**	

Course Objectives:

1. To apply and relate knowledge of mathematics in computer science.

2. To learn proof theory with propositional calculus and induction.
3. To map and express network problem with trees and graphs

Course Prerequisites: Students should have fundamental mathematical knowledge.

Course Outcome:
Students will be able to:

1. Formulate real world problems into statement forms using sets and relations which can be solved or proved mathematically using set theory and logic.
2. Find and map relation between mathematical statements.
3. Design mathematical model from theoretical statements.
4. Find optimum solution using theory of probability.
5. Apply knowledge graphs to solve network problems.
6. Design searching algorithm efficiently by applying tree and tree traversal logic.

UNIT-I **Propositional Logic and Proof Theory** **(06 Hours)**

Sets, Set operations, Finite and Infinite sets, Venn diagram, Principle of inclusion and exclusion, Multisets. Propositions, Conditional Propositions, Logical Connectivity, Propositional calculus, Universal and Existential Quantifiers, Normal forms, methods of proofs.

UNIT-II **Relations and Functions** **(06 Hours)**

Properties of Binary Relations, Closure of relations, Warshall's algorithm, Equivalence, Relations and partitions, Partial ordering relations and lattices, Chains and Anti chains. Functions, Composition of functions, Invertible functions, Pigeonhole Principle.

UNIT-III **Induction and Recurrence Relations** **(06 Hours)**

Mathematical Induction, Linear Recurrence Relations with constant Coefficients, Homogeneous Solutions, Total solutions, Solutions by the method of generating functions

UNIT-IV **Probability** **(06 Hours)**

Basics of permutations and combinations, Discrete Probability, Conditional Probability, Probability distribution: normal, binomial, Poisson, Bernoulli distribution.

UNIT-V **Graphs** **(06 Hours)**

Basic terminology, multi graphs and weighted graphs, paths and circuits, shortest path in weighted graph, Hamiltonian and Euler paths and circuits, factors of a graph, planer graph and Travelling salesman problem.

UNIT-VI **Trees** **(06 Hours)**

Trees, rooted trees, path length in rooted trees, prefix codes, binary search trees, spanning trees and cut set, minimal spanning trees, Kruskal's and Prim's algorithms for minimal Spanning tree.

Assignment List:

1. Write a program to implement following set operations.
   - i) Union
   - ii) Intersection
   - iii) Cartesian product
   - iv) Power set
2. Write a program to implement Warshall's algorithm.
3. Write a program to calculate value of polynomial for variable x.
4. Write a program to find fogoh, where g(x) and h(x) is taken from user.
5. Write a program to check whether Eulerian circuit is present in the given graph.
6. Write a program to find shortest path between the vertices in given graph.
7. Write a program to create binary search tree for the values taken from user.
8. Write a program to implement various tree traversals.
9. Write a program to implement Kruskal's algorithm.
10. Write a program to implement Prim's algorithm.

Text Books:

1) Kenneth H. Rosen, Discrete Mathematics and its Applications, 7th Edition, McGraw Hill.

2) Seymour Lipzschutz, M.Lipson, Discrete Mathematics, 3rd Edition, McGraw Hill.

Reference Books:

1) C. L. Liu, D. P. Mohapatra, Elements of Discrete Mathematics: A Computer Oriented Approach, 4th Edition, McGraw Hill.

2) J.P.Tremblay, R. Manohar, Discrete Mathematical Structures With Applications to Computer Science, McGraw Hill.

Syllabus for Unit Test:

Unit Test -1 **Unit I ,II and III**

Unit Test -2 **Unit IV, V and VI**

Software Project Management

Teaching Scheme	Examination Scheme	Credit Allotted
Theory : 3Hrs/Week	**End Semester Examination: 60 Marks**	**Theory: 3**
	Continuous assessment : 40 Marks	

Course Objectives:

1) To help the students gain understanding of the functions and responsibilities of the manager, and enable them to analyze and understand the environment of the organization.
2) To introduce them with techniques used in the performance of managerial job
3) Enable them to analyze and understand the environment of an IT organization

Course Prerequisites: N/ A

Course Outcome:
Students will be able to:

1) Understand basic concepts of management functions
2) Understand the process of IT project initiation
3) Understand the IT project planning process
4) Understand the team dynamics of a project team
5) Understand the concepts of quality and process improvement for IT projects
6) Introduce modern concepts in IT management

UNIT-I	Conceptual difference between terms Management, Administration and Organization, Functions and Principles of Management, Levels of Management, Type of business organization , Organization structures.	**(06 Hours)**
UNIT-II	Defining Project management life cycle, Gathering and establishing project requirements, Defining the project goals and Scope management, Risk management, Budgeting a project, Creating a work breakdown structure.	**(06 Hours)**
UNIT-III	Building project plan, Preparing and implementing the project plan, Project schedule, Project network diagram creation and analysis, Project constraints, Tracking project progress and financial obligations, Revising the project plan, Establishing change control, Coping with project delays	**(06Hours)**
UNIT-IV	Recruitment and selection, Training, Creating roles and responsibilities, Team Management: Leading, Mechanics, Meetings, Maintaining, Motivating, Conflict Management, Job evaluation and merit rating	**(06 Hours)**
UNIT-V	Metric Frameworks for software projects, Metrics for process and product quality, Quality of deliverables, Quality assurance and standards, Quality planning and control, Process: Classification, measurement, analysis and modeling, Process change, Six sigma, CMM, CMMI, PCMM, ISO standards.	**(06 Hours)**
UNIT-VI	Knowledge management: Definition, needs, techniques and architecture. Learning Organizations, Knowledge management system life cycle, Knowledge workers and knowledge audits, Supply chain management, Change management, Stress management, Credit rating of software projects, Intellectual property rights and Cyber laws	**(06 Hours)**

Assignment List: N/ A

Text Books:

1) Joseph Phillips, "IT Project Management", Tata McGraw-Hill 2003 Edition
2) Pankaj Jalote, "Software Project Management", Addison-Wesley, 2002

Reference Books:

1) Kathy Schwalbe, "Information Technology Project Management", Cengage Learning, 7th Edition

2) Rajib Mall, "Fundamentals of Software Engineering", PHI Learning, 2009, 3rd Edition

3) Ian Sommerville, "Software Engineering", Pearson Education India

4) Roger S. Pressman, "Software Engineering: A Practitioner's Approach",Palgrave Macmillan

5) Elias M. Awad Hassan M. Ghaziri, "Knowledge Management" , Pearson Education

6) By Harold R. Kerzner, "Project Management: A Systems Approach to Planning, Scheduling, and Controlling", John Wiley & Sons Inc.,10th Edition

7) Debora J. Halbert, "Resisting Intellectual Property", Taylor and Francis Group, Routledge-2007

Syllabus for Unit Test:

Unit Test -1 Unit I ,II and III

Unit Test -2 Unit IV,V and VI

Data Structures and Files

Teaching Scheme	Examination Scheme		Credit Allotted	
Theory : 3 Hrs/Week	**End Semester Examination**	**: 60 Marks**	**Theory**	**: 3**
Practical : 2 Hrs/Week	**Continuous Assessment**	**: 40 Marks**	**Tw & Pr**	**: 1**
	Term Work and Practical (Tw & Pr)	**: 50 Marks**		

Course Objectives:

1. To study fundamentals of data stuctures
2. To implement linear sequential and linked organization data structures
3. To study fundamentals of Files and hashing

Course Prerequisites:

Students should have knowledge of

1) 'C' programming
2) Basics of OOP

Course Outcome:

Students will be able to:

1) Understand the fundamentals of data structure.
2) Implement linear sequential data structures.
3) Implement linear linked orgnization data structures.
4) Implement non-linear linked orgnization data structures.
5) Implement searching, sorting techniques.
6) Understand Hashing terms and techniques.

UNIT-I **Review of 'C' programming** **(06 Hours)**

Arrays, Pointers, Structure, Functions, Recursive Function

Introduction to Data Structures:

Concept of Data object, Data structure, Abstract Data Types (ADT), realization of ADT in 'C'. Types of data structures. Algorithm Analysis: Definition and Characteristics of Algorithm, Analyzing Programs, Time and Space Complexity, Big 'O' Notation, Graphical Representation of Time Complexity, best, Average and Worst Case of Complexity

UNIT-II **Linear Data Structures using Sequential Organization:** **(06 Hours)**

Concept of sequential organization, arrays as ADT, sparse matrix, Polynomial representation using array.

Stack: Concepts, Operations on Stacks, Multi-stack, Application of Stack: Polish notation (infix, prefix, postfix expressions), Conversion and Evaluation of expressions

Queue: Concept, Operations on Queue, Circular Queue, Priority Queue, Double Ended Queue, Applications of Queue

UNIT-III **Linear Data Structure Using Linked Organization:** **(06 Hours)**

Linear Data Structures using Linked Organization, Limitations of static memory allocation, Dynamic memory allocation in C. Single Linked List, Double Linked List, Circular Linked List, Generalized Linked List, Application of DLL in dynamic storage management, garbage collection and compaction

UNIT-IV **Non-Linear Data Structure:** **(06 Hours)**

Trees: Basic terminology, Binary Trees, representation and operatations of binary tree, Binary tree traversal (Inorder, Postorder, Preorder), Threaded Binary Tree, Binary Search Tree (Weighted BST), AVL Tree

Graphs: Basic terminology, Representation of Graph using adjacency Matrix, List and Multilist, Graph Traversal (DFS & BFS), Spanning Tree. Kruskal's and Prim's Algorithm for MST, Dijkstra's algorithm for shortest Path.

UNIT-V **Sorting and Searching Techniques:** **(06 Hours)**

Sorting: Need of Sorting and Searching, Internal & External sorting. Bubble sort, Selection sort, Insertion sort, Shell sort, Radix sort, Quick sort, Heap sort, Merge sort. Analysis of sorting techniques,

Searching: Sequential search, Binary search, Fibonacci search

UNIT-VI **File Organization and Hashing:** **(6 Hours)**

Introduction to files, File Structure: Concepts of fields, records and files, Sequential, Indexed and Relative/Random access files, File Organization, Indexing

Hashing: symbol table, Hash tables, Hashing Functions, Overflow Handling and Collision Resolution strategies

Assignment List:

1. Write a Program to implement ibonacci series, factorial of no and checking for prime no.
2. Write a Program to implement functions for Stack, Queue and Circular Queue data structure.
3. Write a Program to convert expression from
    1. Infix to Prefix
    2. Infix to Postfix
4. Write a Program to implement polynomial operations
5. Write a Program to implement Sparse Matrix operations

6. Write a menu driven program to implement Singly Linked List for basic operations
7. Write a menu driven program that implements Doubly Linked List for basic operations
8. Write a Program to implement Binary Search Tree and Traversal in BST(Inorder, Preorder, Postorder)
9. Write a Program to implement Threaded Binary Tree and its Traversals.
10. Write a Program to implement Breadth First search and Depth First Search in graph.
11. Write a C Program to implement Linear and Binary Search
12. Write a Program to implement sorting methods. (Bubble sort, Selection sort, Insertion sort, Quick sort)

Text Books:

1. S. Lipschutz, "Data Structures", McGraw Hill Pub.
2. Y. Langsm, M. Augentin, A. Tanenbaum, "Data Structure Using C and C++", Pearson Education
3. R. Gilberg, B. Forozon, "Data Structure: A pseudo code approach with C"

Reference Books:

1. Ellis Horowitz, Sartaj Sahni , "Data Structures,Algorithms&Applications Inc++, University Press", 2nd Edition
2. Trembley Jean Paul, Sorn Soon Paul G, "An Introducton to Data Structures with Applications", Tata McGraw-Hill Publishing Company, 2008

Syllabus for Unit Test:

Unit Test -1 **Unit I ,II and III**

Unit Test -2 **Unit IV, V and VI**

Platform Independent Programming Paradigm

Teaching Scheme	Examination Scheme		Credit Allotted	
Theory	**: 3 Hrs/ Week**	**End Semester Examination**	**: 60 Marks**	**Theory : 3**
Practical	**: 2 Hrs/ week**	**Continuous assessment**	**: 40 Marks**	**Tw & Or :1**
		Term Work and Oral (Tw & Or)	**: 50 Marks**	

Course Objectives:

1) To provide an understanding of Platform Independent Programming
2) To instill basics of Ruby, Python and Web-applications

Course Prerequisites:

Students should have knowledge of

1) C, C++
2) Basic Knowledge of Computing terminologies

Course Outcome:

Students will be able to:

1) Convey basic concepts of cross platform software development.
2) Analyze programs in Ruby and Python.
3) Demonstrate JAVA concepts in terms of OOP.
4) Analyze the concepts of HTML and CSS for creating webpages.
5) Analyze the applications of Cross Platform Programming.
6) Describe the architecture of JUCE.

UNIT-I	**Introduction** Cross platform software development, Software Platforms, Operating Systems – introduction and its relevance to application software, Scripting, Compilers and Interpreters.	**(06 Hours)**
UNIT-II	**Ruby Programming Language** Semantics, Syntax, data types – strings & collections, conditional statements and loops, Implementation of Class.	**(06 Hours)**
UNIT-III	**Java** Architecture, JVM, Byte code, data types, conditional statements and loops, functions.	**(06 Hours)**
UNIT-IV	**Python** Semantics, Syntax, data types, statements, methods.	**(06 Hours)**
UNIT-V	**Internet** Web servers, Browsers, Webpages, Introduction to Scripting languages, Basics of HTML and CSS.	**(06 Hours)**
UNIT-VI	**Applications** Cross platform development & challenges, Cross platform mobile development, HTML5. **JUCE** Introduction, JUCE Module Format, Introjucer, Data Structure, Working with Media Files.	**(06 Hours)**

Assignment List: (Term work shall consist of Six assignments from above syllabus.)

1) Demonstrate the programming model of Ruby using a simple example.
2) Discuss OOP features available in JAVA.
3) Summarize atleast 10 Methods of Python.
4) Write a Case Study of JUCE Module Format.
5) Explain the various types of CSS with suitable example.
6) Compare HTML webpage with HTML5 webpage.

Text Books:

1) "System Software and Operating System" – D M Dhamdhere (Tata McGraw Hill)
2) "The Ruby Programming Language" - David Flanagan & Yukihiro Matsumoto (O'Reilly Media)

3) "Java - The Complete Reference" - Herbert Schildt(McGraw Hill)
4) "Think Python" - Allen Downey (O'Reilly)
5) "Web Technologies" – Black Book (Dreamtech Press)
6) "Getting started with JUCE" - Martin Robinson (PACKT Publishing)

Reference Books:

1) "Professional Cross-Platform Mobile Development" - Scott Olson, John Hunter(Wrox Publication)

Syllabus for Unit Test:

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

ITL - I

Teaching Scheme	**Examination Scheme**	**Credit Allotted**
Practical : 4 Hrs/Week	**Practical and Term Work : 50 Marks**	**Tw & Pr : 2**

Course Objectives: To

1) Apply concepts of programming language to meet the requirements specified.
2) Sketch an outline of a website with GUI.
3) Solve various problems during development of website.
4) Analyze the given requirement to design the pages for a website.
5) Test the functionalities required.
6) Design web application on an internet.

Course Prerequisites:

Students should have knowledge of

1) Programming language
2) Presentation layer, properties

Course Outcome:

Students will be able to:

1) Design the layout of a website
2) Maintain the presentation logic and business logic
3) Solve customers' requirement by designing web pages
4) Understand principals of GUI
5) Validate the component's role and functionalities associated with it

6) Design website using various client side and server side scripting.

UNIT-I **Introduction to an Internet**: Introduction to internet and its applications, E- mail, telnet, FTP, e-commerce, video conferencing, e-business. Internet service providers, domain name server, World Wide Web and its evolution, uniform resource locator (URL), browsers - internet explorer, netscape navigator, opera, firefox, chrome, mozilla. Search engine, web saver - apache, IIS, proxy server, HTTP protocol. **(06 Hours)**

UNIT-II **HTML** : HTML Tag Reference, Global Attributes, Event Handlers, Document Structure Tags, Formatting Tags, Text Level formatting, Block Level formatting, List Tags, Hyperlink tags, Image and Image maps, Table tags, Form Tags, Frame Tags, Executable content tags. Imagemaps : What are Imagemaps Client-side Imagemaps, Server-side Imagemaps, Using Server-side and Client-side Imagempas together, alternative text for Imagemaps, Tables : Introduction to HTML tables and their structure, The table tags, Alignment, Aligning Entire Table, Alignment within a row, Alignment within a cell, Attributes, Content Summary, Background color, Adding a Caption, Setting the width, Adding a border, Spacing within a cell, Spacing between the cells, spanning multiple rows or columns, Elements that can be placed in a table. Passing form data Style Sheets : What are style sheets, Why are style sheets valuable Different approaches to style sheets, Using Multiple approaches, Linking to style information in s separate file, Setting up style information, Using the tag, embedded style information. **(06 Hours)**

UNIT-III **JavaSript:** Introduction, Client-Side JavaScript, Server-Side JavaScript, JavaScript Objects, JavaScript Security, Operators : Assignment Operators, Comparison Operators Defining Event Handlers, event, onAbort, onBlur, onChange, onClick, onDblClick, onDragDrop, onError, onFocus, onKeyDown, onKeyPress, onKeyUp, onLoad, onMouseDown, onMouseMove, onMouseOut, onMouseOver, onMouseUp, onMove, onReset, onResize, onSelect, onSubmit, onUnload. **(06 Hours)**

UNIT-IV **XML** : Introduction to XML, Anatomy of an XML, document, Creating XML Documents, Creating XML DTDs, XML Schemas, **(06 Hours)**

XSL.

UNIT-V **PHP** : Why PHP and MySQL?, Server-side web scripting, Installing PHP, Adding PHP to HTML, Syntax and Variables, Passing information between pages, Strings, Arrays and Array Functions, Numbers, Basic PHP errors / problems. **(06 Hours)**

UNIT-VI **Advanced PHP and MySQL** : PHP/MySQL Functions, Displaying queries in tables, Building Forms from queries, Basic CRUD functionalities using PHP, Cookies and HTTP, Type and Type Conversions, E-Mail. **(06 Hours)**

Assignment List:

1) Design a web page for Department of Information Technology, BVUCOE, Pune.
2) Develop a website using CSS alignment.
3) Simulate e-album of images using Imagemap.
4) Maintain database of student using XML and publish the data on a web.
5) Using Java Script design a web page that prints factorial / Fibonacci series / any given series.
6) Design a form and validate all the controls placed on the form using Java Script.
7) Design a DTD, corresponding XML document and display it in browser using CSS.
8) Develop MIS for student, faculty, lab and syllabus.
9) Simulate cookies using PHP.
10) Implement CRUD operation on MySQL

Text Book:

1) Web Design The complete Reference, Thomas Powell, Tata McGrawHill

2) PHP : The Complete Reference By Steven Holzner, Tata McGrawHill

Reference Books:

2) HTML and XHTML The complete Reference, Thomas Powell, Tata McGrawHill
3) JavaScript 2.0 : The Complete Reference, Second Edition by Thomas Powell and Fritz Schneider

Advanced Data Structure

Teaching Scheme		Examination Scheme		Credit Allotted	
Theory	: 3 Hrs/Week	End Semester Examination	: 60 Marks	Theory	: 3
Practical	: 2 Hrs/Week	Continuous assessment	: 40 Marks	Tw & Pr	: 1

Term Work and Practical (Tw & Pr) : 50 Marks

Course Objectives:

1)Apply important algorithmic design paradigms and methods of analysis
2)Analyze the Complexity of Algorithms

Course Prerequisites:

Students should have knowledge of basic data structures with their operations.

Course Outcome:

Students will be able to:

1)Understand the analysis of algorithms ,Concept of Complexity of Algorithms
2) Understand and analyze threaded binary trees.
3)Understand graph algorithms
4) Understand pattern matching algorithms
5) Understand queues and its types.
6) Understand dynamic programming

UNIT-I **Analysis of Algorithms** : **(06 Hours)**

Algorithms, Designing Algorithms, Analyzing Algorithms ,frequency count and its importance in analysis of an algorithm, Analysis of algorithm efficiency- symptotic notations and its properties. Performance analysis: Time complexity and Space complexity. Analyzing Recursive Programs.

UNIT-II **Threaded Trees :** **(06 Hours)**

Properties of Threaded trees, Insertion, deletion and traversal AVL Trees, Properties of AVL trees, rotations, insertion and deletion, Red-Black Trees, Properties of Red-Black Trees, rotations, insertion and deletion, B-Trees, Definition of B-Trees, Basic operations on B-Tree, Deleting a key from B tree.

UNIT-III **Graphs:** **(06 Hours)**

Elementary Graph Algorithms: Representations of Graphs – Breadth-First Search – Depth-First Search – Topological Sort – Strongly Connected Components- Minimum Spanning Trees: Growing a Minimum Spanning Tree – Kruskal and Prim- Single-Source Shortest Paths: The Bellman-Ford algorithm – Single-Source Shortest paths in Directed Acyclic Graphs – Dijkstra's Algorithm; All-

Pairs Shortest Paths: Shortest Paths and Matrix Multiplication – The Floyd-Warshall Algorithm.

UNIT-IV **Strings :** **(06 Hours)**

String Matching: The Native String-Matching Algorithm – Brute force, Rabin-Karp and Knuth-Morris-Pratt Algorithms, Boyer-Mooree algorithm, String matching with automata.

UNIT-V **Priority Queues :** **(06 Hours)**

Definition, ADT, Realizing a Priority Queue using Heaps, Definition, insertion, Deletion, External sorting Model, Multiway merge, Polyphase merge.

UNIT-VI **Dynamic Programming :** **(06 Hours)**

Concept of Dynamic Programming, Dynamic Programming basic strategy, General Method of Dynamic Programming, Limitations of Dynamic Programming, traveling salesman problem ,0/1 Knapsack Problem.

Assignment List:

1) To analyze the algorithm of anagram detection by writing a Boolean function that will take two strings and return whether they are anagrams.

2) Given a list of numbers in random order, write an algorithm that works in $O(n\log(n))$ to find the kth smallest number in the list.

3) Implement the non-recursive pre-order tree traversal algorithm.

4) Construct AVL tree for the list {J,F,M,A,N,K,L,A,S,O,P,D}?

5) To implement the Floyd Warshall algorithm to find the shortest path.

6) Given two strings – a text and a pattern, determine whether the pattern appears in the text using Rabin-Karp and Knuth-Morris-Pratt Algorithms.

7) Write a C program to implement queue ADT using Arrays.

8) Write a C program to implement queue ADT using Linked List.

9) To generate the sets S^i, $0 <= i <=3$ for the following knapsack instance: n =3 , $(w_1,w_2,w_3)=(2,3,4)$,$(p_1,p_2,p_3)=(1,2,5)$ and M=6.In addition find an optimal solution.

Text Books:

1) Anany Levitin-"Introduction to design and analysis of algorithms", Third Edition, Pearson Education,2012

2) Aho Hop croft Ullman ―Data Structures and Algorithms, Pearson Education, 2002.

3) "Fundamentals of Computer Algorithms", Horowitz, Sahani, Rajsekharan, Galgotia Publications.

Reference Books:

1) Tanenbaum A.S, Langram Y, Augustine M.J., Data Structures using C & C++‖, Prentice Hall of India, 2002.

2) Mark Allen Weiss, ―Data Structures and Algorithm Analysis in C++‖, Pearson Education, 2002.

3) Data structures and Algorithms in C++, Michael T.Goodrich, R.Tamassia and Mount, Wiley student edition, John Wiley and Sons.

4) Thomas H.Corman,Charles E.Leiserson,Ronald L.Rivest and Clifford Stein,"Introduction to algorithms"Third Edition ,PHI Learning Private Limited,2012**.**

5) Fundamentals of DATA STRUCTURES in C: 2nd ed, Horowitz, Sahni, Anderson-freed, Universities Press.

6) Advanced Data Structures :A.A.Puntambekar,Technical Publications

Syllabus for Unit Test:

Unit Test -1 **Unit I ,II and III**

Unit Test -2 **Unit IV, V and VI**

Digital Electronics and Logic Design

Teaching Scheme	Examination Scheme		Credit Allotted
Theory :3 Hrs/Week	End Semester Examination	: 60 Marks	Theory :3
	Continuous assessment	: 40 Marks	

Course Objectives:

1) To introduce number systems and codes.
2) To outline the formal procedures for the analysis and design of combinational circuits and sequential circuits.
3) To introduce the concept of memories, programmable logic devices and digital ICs.
4) Give students the basic tools for the design and implementation of digital modules and subsystems.

5) Give students the concept of digital logic design.
6) Reinforce theory and techniques taught in the classroom through project assignments.

Course Prerequisites:
Students should have knowledge of
1) Basic electronics

Course Outcome:
Students will be able to:
1) Understand various logic families, number systems and different theorems in Boolean Algebra.
2) Understand various combinational circuits and reduction of Boolean expressions using different techniques.
3) Understand various sequential circuits and solve design problems using this circuits.
4) Understand concept of state machines.
5) Understand different types of memories, working of cache and different advanced DRAMs.
6) Understand the concept and writing programs in VHDL.

UNIT-I Classification of logic families, Characteristics of digital ICs-Speed of operation, power dissipation, figure of merit, fan in, fan out, current and voltage parameters, noise immunity, operating temperatures and power supply requirements.TTL logic. Operation of TTL NAND gate, active pull up, wired AND, open collector output, unconnected inputs. Tri-State logic. CMOS logic – CMOS inverter, NAND, NOR gates, unconnected inputs, wired logic , open drain output. Interfacing CMOS and TTL. Comparison table of Characteristics of TTL, CMOS, ECL, RTL, I2L, DCTL. **(06 Hours)**

Boolean Algebra : Number System : Binary, Hexadecimal numbers, octal numbers and number conversion. Signed Binary number representation: Signed Magnitude, 1's complement and 2's complement representation, Binary, Octal, Hexadecimal Arithmetic: 2's complement arithmetic.Algebra for logic circuits: Logic variables, Logic functions -NOT, AND, NOR, XOR, OR, XNOR, NAND.

UNIT-II Standard representations for logic functions, k map representation of logic functions (SOP m POS forms), minimization of logical functions for min-terms and max-terms (upto 4 variables), don't care conditions, **(06 Hours)**

Design Examples: Arithmetic Circuits, BCD - to – 7 segment decoder, Code converters. Adders and their use as subtractions, look ahead carry, ALU, Digital Comparator, Parity generators/checkers, Multiplexers and their use in combinational logic designs, multiplexer trees, Demultiplexers and their use in combinational logic designs, Decoders, demultiplexer trees. Introduction to Quine McCluskey method.

UNIT-III 1 Bit Memory Cell, Clocked SR, JK, MS J-K flip flop, D and T flip-flops. Use of preset and clear terminals, Excitation Table for flip flops. Conversion of flip flops. Application of Flip flops: Registers, Shift registers, Counters (ring counters, twisted ring counters), Sequence Generators, ripple counters, up/down counters, synchronous counters, lock out, Clock Skew, Clock jitter. Effect on synchronous designs. **(06 Hours)**

UNIT-IV Basic design steps- State diagram, State table, State reduction, State assignment, Mealy and Moore machines representation, Implementation, finite state machine implementation, Sequence detector. **(06 Hours)**

UNIT-V Semiconductor memories: memory organization and operation, expanding memory size, Classification and characteristics of memories, RAM, ROM, EPROM, EEPROM, NVRAM, SRAM, DRAM, expanding memory size, Synchronous DRAM (SDRAM), Double Data Rate SDRAM, Synchronous SRAM, DDR and QDR SRAM, Content Addressable Memory. **(06 Hours)**

UNIT-VI Algorithmic State Machines: ASM charts, notations, design of simple controller, multiplexer controller method:
Introduction to HDL, VHDL: Library, Entity, Architecture, Modeling styles, Data objects, Concurrent and sequential statements, Design examples using VHDL for basic combinational and sequential circuits. **(06 Hours)**

Text Books:

1) Morris Mano, Digital Design, Prentice Hall of India, 2002.
2) R.P. Jain, “Modern Digital Electronics”, 3rd Edition, Tata McGraw-Hill, 2003, ISBN 0 –

07 – 049492 – 4
3) Malvino, D.Leach " Digital Principles and Applications", 5th edition, Tata McGraw Hill

Reference Books:

1) John M.Yarbrough, 'Digital Logic, Application & Design', Thomson, 2002
2) Thomson, 2002. 2. Thomas L.Floyd, "Digital Fundamentals", PHI, 2003.
3) J. Bhaskar, "VHDL Primer" 3rd Edition.PHI Publication

Syllabus for Unit Test:

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

Database Management System

Teaching Scheme	Examination Scheme		Credit Allotted	
Theory : 3Hrs/Week	**Semester Examination**	**: 60 Marks**	**Theory**	**: 3**
Practical : 2Hrs/Week	**Continuous assessment**	**: 40 Marks**	**Tw & Or**	**: 1**
	Term Work and Oral (Tw & Or)	**: 50 Marks**		

Course Objectives:

1) Identify various techniques to communicate with database.
2) Relate relevant data for effective processing of data.
3) Construct a database to maintain data adroitly.
4) Study various queries and tools to deal with the data.
5) Understand the relation between data set and respective means to access it.
6) Understand influence of data in the effective development of software.

Course Prerequisites:

Students should have knowledge of

1) Basic understanding of data and data structure
2) Basic understanding of programming language

Course Outcome:

Students will be able to:

1) Design database to store data related with application.
2) Identify technique to deal with data.
3) Extend power of SQL by adding programming paradigm.
4) Predict suitable environment for data processing as per type data.
5) Apply knowledge of dbms to process the software efficiently.

UNIT-I **Introduction to DBMS:** What is database management system, Use of database system, view of data, relational databases, database architecture, transaction management, Data Models The importance of data models, Basic building blocks, Business rules, The evolution of data models, Degrees of data abstraction. Design of Database, ER-Diagram Database design. ER Model: overview of ER-Model, Constraints, ER-Diagrams, Extended ER Diagrams. **(06 Hours)**

UNIT-II **Relational database model:** Logical view of data, keys, integrity rules. Design of Relational Database: features of good relational database design, Normalization (1NF, 2NF, 3NF, BCNF). Relational Algebra and Calculus Relational algebra: introduction, Selection and projection, set operations, renaming, Joins, Division, syntax, semantics. Operators, grouping and ungrouping, relational comparison. Calculus: Tuple relational calculus, Domain relational Calculus, calculus vs algebra, computational capabilities. **(06 Hours)**

UNIT-III **Integrity Constraints:** What are constraints, types of constrains, Integrity constraints, Views: Introduction to views, data independence, security, updates on views, comparison between tables and views Introduction to SQL: data definition, aggregate function, Null Values, nested sub queries, Joined relations. Triggers. **(06 Hours)**

UNIT-IV **PL/SQL**: Introduction ,Declaring Variables , Writing Executable Statements , Interacting with Oracle Server , Writing Control Structures , Working with Composite Data Types , Writing Explicit Cursors , Writing Implicit Cursors , Handling Exceptions , Creating Procedures , Creating Functions , Managing Subprograms , Creating Packages , More Package concepts , Oracle supplied Packages, Manipulating Large Objects , Creating Database Triggers. **(06 Hours)**

UNIT-V **Transaction management:** ACID properties, serializability and concurrency control, Lock based concurrency control (2PL, Deadlocks), Time stamping methods, optimistic methods, database recovery management. **(06 Hours)**

UNIT-VI **Data Intensive Computing:** Introduction to big data, unstructured data processing using Hadoop , NoSQL database using MangoDB. **(06 Hours)**

Assignment List:

1) Draw an ER Diagram to maintain database of Bank
2) Normalize the database of Library, upto BCNF
3) Perform the following operation for demonstrating the insertion , updation and deletion using the referential integrity constraints
4) Calculate turnover of a banks in pune using group by query
5) WAP to implement autorollback option on deletion using trigger.
6) WAP to implement Procedure to calculate square of a number.

7) Implement implicit cursor using PL/SQL.
8) Simulate two phase locking protocol on the database of Movie.
9) Perform document processing using MangoDB,.
10) Solve word count problem using Hadoop.

Text Books:

1) A Silberschatz, H Korth, S Sudarshan, "Database System and Concepts", Sixth Edition McGraw-Hill
2) Oracle SQL and PL/SQL Guide Till 10gR2
3) Ramkrishna R., Gehrke J., Database Management Systems, 3rd Edition, McGraw-Hill

Reference Books :

1) Rob, Coronel, "Database Systems", Seventh Edition, Cengage Learning.
2) Bipin Desai, Introduction to Database Management Systems.
3) Groff James R., Paul Weinberg, LAN times guide to SQL.

Syllabus for Unit Test:

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

Engineering Mathematics-III

Teaching Scheme:	Examination Scheme:	Credits Allotted:
Theory : 3 Hrs / Week	**End Semester Examination: 60 Marks**	**Theory : 4**
Tutorial : 1 Hrs/Week	**Continuous Assessment : 40 Marks**	

Course Pre-requisites:

Students should have basic knowledge of:

1. Differential calculus
2. Integral calculus
3. Complex numbers
4. Vector algebra

Course Objectives:

To develop ability to use the mathematical techniques, skills, and tools necessary for

engineering practice.

Course Outcomes:

At the end of the course , a student will be able to:

1. Form mathematical modeling of systems using differential equations and ability to solve linear differential equations with constant coefficient.
2. Apply basics of analytic functions and the basics in complex integration which is used to evaluate complicated real integrals.
3. Apply theorems to compute the Laplace transform, inverse Laplace transforms.
4. Solve difference equation by Z-transform.
5. Calculate the gradients and directional derivatives of functions of several variables.
6. Use Green's theorem to evaluate line integrals along simple closed contours on the plane.

UNIT - I **Linear Differential Equations (LDE)** **(06 Hours)**

Solution of nth order LDE with Constant Coefficients, Method of Variation of Parameters, Cauchy's &Legendre's DE, Solution of Simultaneous & Symmetric Simultaneous DE, Modeling of Electrical Circuits.

UNIT - II **Complex Variables** **(06 Hours)**

Functions of Complex Variables, Analytic Functions, C-R Equations, Conformal Mapping, Bilinear Transformation, Cauchy's Theorem, Cauchy's Integral Formula, Laurent's Series, Residue Theorem

(06 Hours)

UNIT - III **Transforms**

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 – V **Laplace Transform (LT)** **(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 - V **Vector Differential Calculus** **(06Hours)**

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** **(06 Hours)**

Line, Surface and Volume integrals, Work-done, Green's Lemma, Gauss's Divergence Theorem, Stoke's Theorem, Applications to Problems in Electro-Magnetic Fields.

Assignments:

1. Linear Differential Equations
2. Complex Variables
3. Transforms
4. Laplace Transform
5. Vector Differential Calculus
6. Vector Integral Calculus

Text Books:

1. Advanced Engineering Mathematics by Peter V. O'Neil (Cengage Learning).
2. Advanced Engineering Mathematics by Erwin Kreyszig (Wiley Eastern Ltd.).

Reference Books:

1. Engineering Mathematics by B.V. Raman (Tata McGraw-Hill).
2. Advanced Engineering Mathematics, 2e, by M. D. Greenberg (Pearson Education).
3. Advanced Engineering Mathematics, Wylie C.R. & Barrett L.C. (McGraw-Hill, Inc.)
4. Higher Engineering Mathematics by B. S. Grewal (Khanna Publication, Delhi).
5. Applied Mathematics (Volumes I and II) by P. N. Wartikar & J. N. Wartikar (Pune Vidyarthi Griha Prakashan, Pune).

Syllabus for Unit Test:

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

Computer Graphics

Teaching Scheme	Examination Scheme	Credit Allotted

Theory	**: 3Hrs/Week**	**End Semester Examination**	**: 60 Marks**	**Theory**	**: 3**
Practical	**: 2Hrs/Week**	**Continuous assessment**	**: 40 Marks**	**Tw &Pr**	**: 1**
		Term Work and Practical(Tw & Pr)	**: 50 Marks**		

Course Objectives:

1) learn basic and fundamental computer graphics techniques;
2) learn image synthesis techniques
3) examine applications of modeling, design and visualization

Course Prerequisites:

Students should have knowledge of

1)C / C++ programming
2)Data structures and files

Course Outcome:

Students will be able to:

1) Understand history and evolution of computer graphics, both hardware and software and use a current graphics API (OpenGL).
2) Implement algorithms including: line drawing, polygon filling, clipping, and transformations.
3) Write programs that demonstrate geometrical 2 D transformations
4) Write programs that demonstrate geometrical 3 D transformations
5) Understand illumination and shading concepts
6) Implement Fractals.

UNIT-I **(06 Hours)**

Introduction to Computer Graphics

Overview of Computer Graphics, , Description of graphics devices, Input Devices, Graphics Devices, Display Technologies, Raster Refresh (Raster-Scan) Graphics Displays, Scan Basics, Video Basics, The Video Controller, Random-Scan, Display Processor, LCD displays, display file and structure, display processor, Graphics file formatComputer Graphics Application and Software Introduction to OPENGL, Applications of Computer graphics, Animation.

UNIT-II **(06 Hours)**

Scan conversion – lines, circles and Ellipses; Filling polygons and clipping algorithms

Line and circle drawing: DDA, Bresenham's, algorithms, Thick line segment, Aliasing, Anti aliasing , Polygons Filling algorithms, edge data structure, Line Clipping algorithms– Cyrus-Beck, Cohen-Sutherland, Polygon Clipping algorithms: Sutherland Hodgeman, Color models.

UNIT-III Transformations: **(06 Hours)**

Matrix Representation of 2D Transformations: Rotation, Reflection, Scaling, Combined Transformation, Translations and Homogeneous Coordinate system ,Transformation of Points, Transformation of the Unit Square, Solid Body Transformations, Rotation About an Arbitrary Point, Reflection through an Arbitrary Line, A Geometric Interpretation of Homogeneous Coordinates, Window-to-Viewport Transformations. Introduction, Matrix Representation of 3D Transformations: Scaling, Shearing, Rotation, Reflection, Translation, Multiple Transformation, Rotation about an Arbitrary Axis in Space, Reflection through an Arbitrary Plane, Composition of 3D Transformations.

UNIT-IV Projections: **(06 Hours)**

Affine and Perspective Geometry, Perspective Transformations, Techniques for Generating Perspective Views, Vanishing Points, the Perspective Geometry and camera models, Orthographic Projections, Axonometric Projections, Oblique Projections, View volumes for projections. Stages in 3D viewing, Canonical View Volume (CVV), Specifying an Arbitrary 3D View, Examples of 3D Viewing, viewing, Coordinate Systems and matrices, camera model and viewing pyramid.

UNIT-V Hidden Surface Determination: **(06 Hours)**

Techniques for efficient Visible-Surface Algorithms, Categories of algorithms, Back face removal, The z-Buffer Algorithm, Scan-line method, Painter's algorithms , Area sub-division method, BSP trees, Visible-Surface Ray Tracing, comparison of the methods.

Illumination and Shading

Illumination and Shading Models for Polygons, Reflectance properties of surfaces, Ambient, Specular and Diffuse reflections, Atmospheric attenutation, Phong's model, Gouraud shading, some examples.

UNIT-VI Curves and fractals: **(06 Hours)**

Curve Representation, Nonparametric Curves, Parametric Curves, The General Conic Equation, Representation of Space Curves, Cubic Splines, Bezier Curves, B-spline Curves, B-spline Curve Subdivision,

Parametric Cubic Curves, Quadric Surfaces. Bezier Surfaces, fractals and fractal surfaces, Hilbert's curve, Koch curve.

Assignment List:

1) Laboratory exercises will normally be conducted using the currently available computer graphics API such as OpenGL

2) Implement Cohen Sutherland/DDA line drawing algorithm.

3) Implement Cohen Sutherland/DDA circle drawing algorithm.

4) Write a program to implement polygon filling algorithm.

5) Implement Cohen Sutherland Line clipping algorithm

6) Implement following 2D Transformations:

- i) Translation
- ii) Rotation
- iii) Scaling
- iv) Shearing

7) Implement 3D Transformations

- i) Translation
- ii) Rotation
- iii) Scaling
- iv) Shearing
- v) Shearing

8) Write a program to draw fractals

9) Write a program to draw Koch curve

10) Write a program to draw Hilbert'scurve

11) Using OPENGL libraries create an animation.

Text Books:

1) S. Harrington, "Computer Graphics", 2nd Edition, McGraw-Hill Publications, 1987, ISBN 0 – 07 – 100472 – 6. 2.

2) D. Rogers, "Procedural Elements for Computer Graphi cs", 2nd Edition, Tata McGraw-Hill Publication, 2001, ISBN 0 – 07 – 047371– 4.

Reference Books:

1) J. D. Foley, A. Van Dam, S. K. Feiner and J. F. Hughes, Computer Graphics - Principles and Practice, Second Edition in C, Pearson Education, 2003.

2) D. Hearn and M. Pauline Baker, Computer Graphics (C Version), Pearson Education, 2nd Edition, 2004.

3) D. F. Rogers and J. A. Adams, Mathematical Elements for Computer Graphics, 2nd Edition, McGraw-Hill International Edition, 1990.

4) 4. F. S. Hill Jr., Computer Graphics using OpenGL, Pearson Education, 2003

Syllabus for Unit Test:
Unit Test -1 Unit I ,II and III
Unit Test -2 Unit IV, V and VI

IT Lab – II

Teaching Scheme	Examination Scheme	Credit Allotted
Practical :4 Hrs/Week	Term work and Practical (Tw & Pr) : 50 Marks	Tw & Pr : 2

Course Objectives: To

1) Compute time and space complexity for given program.
2) Demonstrate concepts OOPS using java
3) Solve specified requirement
4) Infer various approach to decide efficiency of given approach.
5) Formulate given problem by providing the proof of behavior of given model.
6) Design an application using platform independent approach.

Course Prerequisites:
Students should have knowledge of

1) Object Oriented Programming language
2) Logic to solve given problem

Course Outcome:
Students will be able to:

1) Design simple application meeting the requirements.
2) Develop their logical skill through various assignments and practicals.
3) Breakdown complex problem into subpart and then handle every part to achieve the goal.
4) Model a solution to any real world problem
5) Analyze significance of platform independency.

6) Design application using object oriented norms.

UNIT-I **Introduction to Java** : **(06 Hours)**

Java Fundamentals, Features of Java OOPs concepts Java virtual machine Reflection byte codes Byte code interpretation Data types, variable, arrays, expressions, operators, and control structures Objects and classes .

UNIT-II **Classes and objects :** **(06 Hours)**

Java Classes, Abstract classes Static classes Inner classes Packages Wrapper classes Interfaces This Super Access control, embedded style information Exception handling Exception as objects Exception handling mechanism: Try catch finally Throw, throws.

UNIT-III **Object oriented Properties:** **(06 Hours)**

Inheritance, Encapsulation, Polymorphism, Data Binding, data abstraction. Implementation of these concepts using various statements like if, switch and loops like for,do – while, while.

UNIT-IV **IO mechanism:** **(06 Hours)**

IO package Input streams Output streams Object serialization Deserialization Sample programs on IO files Filter and pipe streams

UNIT-V **Threading and Multithreading:** **(06 Hours)**

Lifecycle of Thread, Basic functions of thread, multithreading, synchronization.

UNIT-VI **Collections and Generics:** **(06 Hours)**

Introduction to collection framework, List, Set, Maps**,** utility class, Reflection API.

Assignment List:

1) WAP to crate array of an object to maintain data of an employee.
2) WAP to design user defined exception to reject negative numbers
3) Count the number of objects created for a class using static member function.
4) Write programs on interfaces.
5) Write programs on packages.
6) Write programs to copy contents of file into other file using all possible alternatives.
7) WAP to simulate traffic signal using multithreading
8) WAP to Maintain the record of student using collection API.

9) WAP to map roll and name to maintain data of students.
10) WAP to maintain record of an employee using List.

Text Books:

1) Programming with Java A Primer, E. Balaguruswamy Tata Mc-Graw Hill Companies.
2) SCJP 1.6 – Khalid Mughal
3) SCJP 1.6 – Kathy Sierra

Reference Books

1) Herbert Schildt, Java 2 Complete Reference – 5th Edition, Tata Mc-Gra Hill
2) Dietel & Dietel, Java How to Program