Syllabus

S.E. (Information Technology) 2015 Course

(With effect from Academic Year 2016 - 17)

SAVITRIBAI PHULE PUNE UNIVERSITY

THE SYLLABUS IS PREPARED BY:

B.O.S. in Information Technology, SavitribaiPhule Pune University
PROGRAM EDUCATIONAL OBJECTIVES

The students of Information Technology course after passing out will

1. Possess strong fundamental concepts in mathematics, science, engineering and Technology to address technological challenges.

2. Possess knowledge and skills in the field of Computer Science and Information Technology for analyzing, designing and implementing complex engineering problems of any domain with innovative approaches.

3. Possess an attitude and aptitude for research, entrepreneurship and higher studies in the field of Computer Science and Information Technology.

4. Have commitment to ethical practices, societal contributions through communities and lifelong learning.

5. Possess better communication, presentation, time management and team work skills leading to responsible & competent professionals and will be able to address challenges in the field of IT at global level.
PROGRAM OUTCOMES

The students in the Information Technology course will attain:

1. an ability to apply knowledge of mathematics, computing, science, engineering and technology;

2. an ability to define a problem and provide a systematic solution with the help of conducting experiments, analyzing the problem and interpreting the data;

3. an ability to design, implement, and evaluate a software or a software/hardware system, component, or process to meet desired needs within realistic constraints;

4. an ability to identify, formulate, and provide systematic solutions to complex engineering/Technology problems;

5. an ability to use the techniques, skills, and modern engineering technology tools, standard processes necessary for practice as a IT professional;

6. an ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems with necessary constraints and assumptions;

7. an ability to analyze and provide solution for the local and global impact of information technology on individuals, organizations and society;

8. an ability to understand professional, ethical, legal, security and social issues and responsibilities;

9. an ability to function effectively as an individual or as a team member to accomplish a desired goal(s);

10. an ability to engage in life-long learning and continuing professional development to cope up with fast changes in the technologies/tools with the help of electives, professional organizations and extra-curricular activities;

11. an ability to communicate effectively in engineering community at large by means of effective presentations, report writing, paper publications, demonstrations;

12. an ability to understand engineering, management, financial aspects, performance, optimizations and time complexity necessary for professional practice;

13. an ability to apply design and development principles in the construction of software systems of varying complexity.
**S.E. (Information Technology) 2015 Course to be implemented from June 2016**

**SEMESTER – I**

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>Subject</th>
<th>Teaching Scheme</th>
<th>Examination Scheme</th>
<th>Total Marks</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>214441</td>
<td>Discrete Structures</td>
<td>4 -- --</td>
<td>50 50 -- -- --</td>
<td>100</td>
<td>4</td>
</tr>
<tr>
<td>214442</td>
<td>Computer Organization &amp; Architecture</td>
<td>4 -- --</td>
<td>50 50 -- -- --</td>
<td>100</td>
<td>4</td>
</tr>
<tr>
<td>214443</td>
<td>Digital Electronics and Logic Design</td>
<td>4 -- --</td>
<td>50 50 -- -- --</td>
<td>100</td>
<td>4</td>
</tr>
<tr>
<td>214444</td>
<td>Fundamentals of Data Structures</td>
<td>4 -- --</td>
<td>50 50 -- -- --</td>
<td>100</td>
<td>4</td>
</tr>
<tr>
<td>214445</td>
<td>Problem Solving and Object Oriented Programming</td>
<td>4 -- --</td>
<td>50 50 -- -- --</td>
<td>100</td>
<td>4</td>
</tr>
<tr>
<td>214446</td>
<td>Digital Laboratory</td>
<td>-- -- 2</td>
<td>-- -- 25 50 -- --</td>
<td>75</td>
<td>1</td>
</tr>
<tr>
<td>214447</td>
<td>Programming Laboratory</td>
<td>-- -- 4</td>
<td>-- -- 25 50 -- --</td>
<td>75</td>
<td>2</td>
</tr>
<tr>
<td>214448</td>
<td>Object Oriented Programming Lab.</td>
<td>-- -- 2</td>
<td>-- -- 25 50 -- --</td>
<td>75</td>
<td>1</td>
</tr>
<tr>
<td>214449</td>
<td>Communication Skills</td>
<td>-- -- 2</td>
<td>-- -- 25 -- -- --</td>
<td>25</td>
<td>1</td>
</tr>
<tr>
<td>Audit Course</td>
<td></td>
<td>-- --</td>
<td>-- -- -- -- -- --</td>
<td>--</td>
<td></td>
</tr>
</tbody>
</table>

**Total** | 20 -- 10 250 250 100 150 -- 750 25

**Total of Part-I** | 30 Hours 750

**SEMESTER – II**

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>Subject</th>
<th>Teaching Scheme</th>
<th>Examination Scheme</th>
<th>Total Marks</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>207003</td>
<td>Engineering Mathematics - III</td>
<td>4 1 --</td>
<td>50 50 25 -- -- --</td>
<td>125</td>
<td>5</td>
</tr>
<tr>
<td>214450</td>
<td>Computer Graphics</td>
<td>3 -- --</td>
<td>50 50 -- -- --</td>
<td>100</td>
<td>3</td>
</tr>
<tr>
<td>214451</td>
<td>Processor Architecture &amp; Interfacing</td>
<td>4 -- --</td>
<td>50 50 -- -- --</td>
<td>100</td>
<td>4</td>
</tr>
<tr>
<td>214452</td>
<td>Data Structures &amp; Files</td>
<td>4 -- --</td>
<td>50 50 -- -- --</td>
<td>100</td>
<td>4</td>
</tr>
<tr>
<td>214453</td>
<td>Foundations of Communication &amp; Computer Network</td>
<td>4 -- --</td>
<td>50 50 -- -- --</td>
<td>100</td>
<td>4</td>
</tr>
<tr>
<td>214454</td>
<td>Processor Interfacing Laboratory</td>
<td>-- -- 4</td>
<td>-- -- 25 50 -- --</td>
<td>75</td>
<td>2</td>
</tr>
<tr>
<td>214455</td>
<td>Data Structure &amp; Files Laboratory</td>
<td>-- -- 4</td>
<td>-- -- 25 50 -- --</td>
<td>75</td>
<td>2</td>
</tr>
<tr>
<td>214456</td>
<td>Computer Graphics Laboratory</td>
<td>-- -- 2</td>
<td>-- -- 25 50 -- --</td>
<td>75</td>
<td>1</td>
</tr>
<tr>
<td>Audit Course</td>
<td></td>
<td>-- --</td>
<td>-- -- -- -- -- --</td>
<td>--</td>
<td></td>
</tr>
</tbody>
</table>

**Total** | 19 01 10 250 250 100 150 -- 750 25

**Total of Part-II** | 30 Hours 750
SEMESTER - I
Teaching Scheme:  Credit  Examination Scheme:
Lectures: 4 Hours/Week  04  In-Semester (Online): 50 Marks

Prerequisites: Basic Mathematics

Course Objectives:
1. Learn the use of set, proof techniques and determine logical possibilities in a given situation.
2. Learn relations, functions among various entities in real world.
3. Learn to apply relations and functions in real life.
4. Learn to formulate problem mathematically using graph theory and trees.

Course Outcomes:
By the end of the course, students should be able to
1. Use set, relation and function to formulate a problem and solve it
2. Use graph theory and trees to formulate the problems and solve them
3. Use mathematical propositions and proof techniques to check the truthfulness of a real life situation.

Course Contents

UNIT I  PERMUTATIONA, COMBINATIONS & DISCRETE PROBABILITY  6 Hours

UNIT II SETS AND PROPOSITIONS  6 Hours
Sets, Combinations of sets, Venn Diagrams, Finite and Infinite sets, Uncountable infinite sets, Principle of inclusion and exclusion, multisets. Propositions, Conditional Propositions, Logical Connectivity, Propositional calculus, Universal and Existential Quantifiers, Normal forms, methods of proofs, Mathematical Induction

UNIT III RELATIONS AND FUNCTIONS  6 Hours
Properties of Binary Relations, Closure of relations, Warshall’s algorithm, Equivalence Relations and partitions, Partial ordering relations and lattices, Chains and Anti chains.

Recurrence Relations
Recurrence Relation, Linear Recurrence Relations With constant Coefficients, Homogeneous Solutions, Total solutions, solutions by the method of generating functions

Functions
Functions, Composition of functions, Invertible functions, Pigeonhole Principle, Discrete Numeric functions and Generating functions, Job scheduling Problem.

UNIT IV GRAPH THEORY  6 Hours
Basic terminology, representation of graph in computer memory, multi graphs and weighted graphs, Subgraph, Isomorphic graph, Complete, regular and bipartite graphs, operation on graph, paths and circuits, Hamiltonian and Euler paths and circuits, shortest path in weighted graph (Dijkstra’s algorithm),
factors of a graph, planer graph and Travelling salesman problem, Graph coloring.

UNIT - V   TREES  6 Hours
Trees, rooted trees, path length in rooted trees, prefix codes and optimal prefix codes, binary search trees, tree traversals, spanning trees, Fundamental circuits and cut set, minimal spanning trees, Kruskal's and Prim's algorithms for minimal spanning tree, The Max flow –Min cut theorem (transport network).

UNIT – VI   GROUPS AND RINGS  6 Hours
Algebraic Systems, Groups, Semi Groups, Monoid, Subgroups, Permutation Groups, Codes and Group codes, Isomorphism and Automorphisms, Homomorphism and Normal Subgroups, Ring, Integral Domain, Field, Ring Homomorphism, Polynomial Rings and Cyclic Codes

Text Books

Reference Books
Teaching Scheme: Lectures: 4 Hours/Week
Credits: 04
Examination Scheme:
In-Semester (Online): 50 Marks
End-Semester: 50 Marks

Prerequisites: Fundamental of Programming Languages

Course Objectives:
1. To understand the structure, function & characteristics of computer systems.
2. To understand the design of the various functional units of digital computers.
3. To understand instruction level parallelism & parallel organization of multi-processor & multi-core systems

Course Outcomes:
On completion of the course, learner will be able to–
1. Solve problems based on computer arithmetic.
2. Explain processor structure & its functions.
3. Obtain knowledge about micro-programming of a processor.
4. Understand concepts related to memory & IO organization.
5. Acquire knowledge about instruction level parallelism & parallel organization of multi-processors & multi-core systems.

Course Contents

UNIT – I COMPUTER EVOLUTION, PERFORMANCE MEASUREMENT & ARITHMETIC 8 Hours
Computer Performance Measurement – Benchmarks (SPEC) for Evaluation, Metrics such as CPU Time, Throughput, etc., Aspects & Factors affecting Computer Performance, Comparing Computer Performances, Marketing Metrics – MIPS & MFLOPS, Speedup & Amdahl’s Law
Booths Algorithm For Signed Multiplication & its Hardware Implementation, Restoring And Non Restoring Division Algorithms & its Hardware Implementation

UNIT – II THE CENTRAL PROCESSING UNIT 8 Hours
Arithmetic & Logic Unit.
Instruction Sets: - Machine Instruction Characteristics, Types of Operands and Types of Operations, Addressing Modes, Instruction Formats, Instruction Types
Processor Structure and Function - Processor Organization, Register Organization, The Instruction Cycle and Instruction Pipelining.
RISC: Instruction Execution Characteristics, RISC Vs CISC, RISC Architecture - MIPS.

UNIT – III THE CONTROL UNIT 8 Hours
Instruction Cycle & Micro Operations, Functional Requirements & Operations of the Control Unit, Block Schematic & Control Signals, Single Bus Processor Organization, Control Signal example with Micro Operations and Register Transfer.
Control Unit Design Methods - Hardwired Control – State Table Method, Design example - Multiplier CU.
Micro-Programmed Control - Basic Concepts, Microinstructions & Formats, Control Memory, Micro-
Programmed Control Unit Schematic, Microinstruction Sequencing - Design Considerations, Sequencing Techniques, Address Generation, Microinstruction Execution - A Taxonomy of Microinstructions, Microinstruction Encoding.

UNIT – IV  Memory & I/O Organization  8Hours

UNIT – V  Instruction level Parallelism  8Hours

UNIT - VI  Parallel Organization  8Hours

Text Books

Reference Books
# 214443: DIGITAL ELECTRONICS AND LOGIC DESIGN

**Teaching Scheme:**
- Lectures: 4 Hours/Week
- Credits: 04

**Examination Scheme:**
- In-Semester (Online): 50 Marks
- End-Semester: 50 Marks

**Prerequisites:** Basic Electronics Engineering

**Course Objectives:**
1. To learn and understand basic digital design techniques.
2. To develop design and implementation skills of combinational and sequential logic circuits.
3. To introduce digital logic design software such as VHDL Programming.

**Course Outcomes:**
1. Spectacle an awareness and apply knowledge of number systems, codes, Boolean algebra and use necessary A.C, D.C Loading characteristics as well as functioning while designing with logic gates.
2. Use logic function representation for simplification with K-Maps and analyze as well as design Combinational logic circuits using SSI & MSI chips.
3. Analyze Sequential circuits like Flip-Flops (Truth Table, Excitation table), their conversion & design the applications.
4. Identify the Digital Circuits, Input/Outputs to replace by FPGA
5. Use VHDL programming technique with different modeling styles for any digital circuits.

**Course Contents**

## UNIT – I  NUMBER SYSTEM AND LOGIC FAMILIES  8 Hours
Introduction to digital electronics & Boolean algebra.

**Number Systems** - Binary, Octal, Hexadecimal and their conversions.  
**Signed Binary number representation and Arithmetic’s:** Signed & True Magnitude, 1’s complement, 2’s complement representation and arithmetic’s.  
**Codes:** BCD, Excess-3, Gray code, Binary Code and their conversion.  
Switching characteristics of BJT & FET, IC Characteristics.  
**TTL:** Standard TTL characteristics, Operation of TTL NAND gate, Subfamilies, Configurations-Active pull-up, Wired AND, totem pole, open collector.  
**CMOS:** Standard CMOS characteristics, operation of CMOS NAND, Subfamilies, CMOS configurations Wired Logic, Open drain outputs.  
Comparison of TTL & CMOS, Interfacing: TTL to CMOS and CMOS to TTL

## UNIT – II  COMBINATIONAL LOGIC DESIGN  8 Hours
Logic minimization: Representation of truth-table, SOP form, POS form, Simplification of logical functions, Minimization of SOP and POS forms, don’t care Conditions.  
**Reduction techniques:** K-Maps up to 4 variables and Quine - McClusky technique.  
**Introduction to MSI functions & chips** - Multiplexers (IC 74151 and IC 74153), Decoder / Demultiplexer (IC 74138), Encoder (IC 74147), Binary adder (IC 7483).
CLC design using MSI chips – BCD & Excess 3 adder & subtractor using IC 7483, Implementation of logic functions using IC 74151, 74153 & 74138.

UNIT – III SEQUENTIAL LOGIC 8 Hours
Introduction to sequential circuits. Difference between combinational circuits and sequential circuits, memory element – latch.
Flip- Flops: Design, truth table, excitation table of SR, JK, D, T flip flops. Study of flip flops with asynchronous and synchronous Preset & Clear, Master Slave configuration, conversion from one type to another type of flip flop. Study of flip flop ICs - 7473, 7474, 7476.
Application of flip-flops – Bounce elimination switch, Counters- asynchronous, synchronous and modulo counters study of modulus n counter ICs- 7490, 74191 & their applications to implement mod counters.

UNIT – IV SEQUENTIAL LOGIC DESIGN 8 Hours
Registers- Buffer register, shift register types - SISO, SIPO, PISO & PIPO, applications of shift registers - ring counter, twisted ring counter, study of universal shift register IC – 74194,
Sequence generators using counters & shift register, Pseudo Random Binary Sequence Generator.
Basic design steps- State diagram, State table, State reduction, State assignment, Mealy and Moore machines representation, Implementation, finite state machine implementation, sequence detector using Moore & Mealy model.

UNIT – V PROGRAMMABLE LOGIC DEVICES AND INTRODUCTION TO HDL 6 Hours
Algorithmic State Machines- ASM notations, charts (eg- counters, washing machine, lift controller, vending machine), design using multiplexer controller method (eg- counters).
Introduction to PLD’s – ROM, PAL, PLA, Design of 4 variable SOP using PLDs, Basic architecture of SPLD and CPLD, Study of CPLD architecture XC9572, Basic architecture of FPGA, CPLD. Design flow (Basic Concept of Simulation and Synthesis)
Introduction to HDL – Necessity, Characteristics & Types.

UNIT - VI VHDL PROGRAMMING 6 Hours
Introduction to VHDL - Library, Package, Entity, Architecture, Data Objects (Variable, signal & constant), Data Types (scalar, composite array type & predefined data types, Attributes (necessity and use. ‘event attribute). VHDL Modeling styles – Dataflow, behavioral & structural
VHDL statements - Concurrent Statements (With. Select, When..Else), Sequential Statements (if..else, case)
VHDL design Examples - Multiplexer, binary adder, counter, shift register.

Text Books

Reference Books
214444 : FUNDAMENTAL OF DATA STRUCTURES

Teaching Scheme:  
Lectures: 4 Hours/Week  
Credits: 04

Examination Scheme:  
In-Semester (Online): 50 Marks  
End-Semester: 50 Marks

Prerequisites: Fundamental knowledge of ‘C’ and basics of algorithms

Course Objectives:
1. To learn C language constructs and pointers in depth.
2. To learn algorithm development and analysis of algorithms.
3. To learn linear data structures and their applications
4. To learn different searching and sorting techniques

Course Outcomes:
1. Student will be able to apply appropriate constructs of C language, coding standards for application development.
2. Students will be to use dynamic memory allocation concepts and file handling in various application developments.
3. Students will be able to perform basic analysis of algorithms with respect to time and space complexity
4. Students will be able to select appropriate searching and/or sorting techniques in the application development
5. Students will be able to select and use appropriate data structures for problem solving and programming
6. Students will be able to use algorithmic foundations for solving problems and programming

Course Contents

UNIT – I  C BASICS  6 Hours
Control structures, arrays, functions and parameter passing Structure and Union, String manipulation, matrix operations.

UNIT – II  POINTERS IN C AND FILE HANDLING  9 Hours
Introduction to Pointers, dynamic memory allocation, pointer to pointer, pointer to single and multidimensional arrays, array of pointers, string and structure manipulation using pointers, pointer to functions. Pointer to file structure and basic operations on file, file handling in C.

UNIT – III  INTRODUCTION TO DATA STRUCTURES AND ANALYSIS OF ALGORITHMS  5 Hours
Introduction to Data Structures: Concept of data, Data object, Data structure, Abstract Data Types, realization of ADT in ‘C’. Concept of Primitive and non-primitive, linear and Non-linear, static and dynamic, persistent and ephemeral data structures.
UNIT – IV    SEARCHING AND SORTING TECHNIQUES       7 Hours
Need of searching and sorting, Concept of internal and external sorting, sort stability. Searching
methods: Linear and binary search algorithms their comparison and complexity analysis
Sorting methods: Bubble, selection, insertion, merge, quick, bucket sort and their time and space
complexity analysis

UNIT – V    LINEAR DATA STRUCTURES USING SEQUENTIAL ORGANIZATION        8 Hours
Concept of sequential organization, Concept of Linear data structures, Concept of ordered list,
Multidimensional arrays and their storage representation: row major and column major form and
address calculation. Representation of sparse matrix using arrays, algorithms for sparse matrix
addition, simple and fast transpose, polynomial representation using arrays. Analysis of these
algorithms. Introduction to Stack and Queue, and their implementation using sequential organization,
use of stack in recursion.

UNIT – VI    LINEAR DATA STRUCTURES USING LINKED ORGANIZATION        8 Hours
Concept of linked organization, singly linked list, doubly linked list, circular linked list. Linked list as an
ADT. Representation and manipulations of polynomials using linked lists, comparison of a sequential
and linked memory organization, concept of Generalized Linked List, representation polynomial using
GLL.

Text Books

2. R. Gilberg, B. Forouzan, "Data Structures: A pseudo code approach with C", Cenage Learning,
   ISBN 9788131503140.

Reference Books

   Second Edition
4. Seymour Lipschutz, “Data structures with C”, Schaum’s Publication
5. Aaron Tanenbaum, “Data Structures using C”, Pearson Education
Prerequisites: Principles of Programming Languages, Fundamentals of Data Structures

Course Objectives:
1. Employ a problem-solving strategy to breakdown a complex problem into a series of simpler tasks.
2. Execute problem-solving actions appropriate to completing a variety of sub problems.
3. Apply analytical and logical thinking to extract facts from a problem description and determine how they relate to one another and to the problems to be solved.
4. Design and implement an object oriented solution to solve a real life problem.
5. Develop problem-solving and programming skills using OOP concept.

Course Outcomes:
After studying this subject student should be able to
1. Break a problem into logical pieces and develop algorithms for solving simple problems.
2. Abstract data and entities from the problem domain, build object models and design software solutions using object-oriented principles and strategies.
4. Develop programs that appropriately utilize key object-oriented concepts.

Course Contents

UNIT – I  Problem Solving Concepts  6 Hours
General Problem Solving Concepts-Types of problems, problems solving with computers, difficulties with problem solving, Problem Solving Aspects, Problem Solving Concepts for computer- constants and variables, data types, functions, operators, expressions and equations, Programming Concepts – communicating with computers, organizing the problem, using the tools, testing the solution, coding the program, Top down design

UNIT – II  Problem Solving with Logic Structures  6 Hours
Programming Structure - modules and their functions, cohesion & Coupling, Local and global variable, parameters, return values, variable names and data dictionaries, four logic structures. Problem solving with sequential logic structure - The sequential logic structure, solution development. Problem Solving with Decisions – decision logic structure, multiple if/then/else instructions, straight-through logic, positive logic, negative logic, logic conversion, decision tables. Problem solving with loops and case logic structures

UNIT – III  Foundations of Object Oriented Programming  6 Hours
Introduction: Introduction to procedural, modular, object-oriented and generic programming techniques, Limitations of procedural programming, Need of object-oriented programming, fundamentals of object-oriented programming: objects, classes, data members, methods, messages, data encapsulation, data abstraction and information hiding, inheritance, polymorphism
++ Extensions to C : Variable declarations, global scope, ‘const’, reference variables, operators in C++(scope resolution, new, delete), dynamic memory allocation, function prototypes, default and constant arguments, ‘cin’, ‘cout’, inline functions
Class: Defining a class, data members and member functions, public, private and protected members, inline member functions, static data members, static member functions, constructors, destructors, array of objects, classes, objects and memory, class as ADTs and code reuse

UNIT – IV Overloading and Inheritance 8 Hours
Function overloading, friend function, friend class
Operator Overloading: Introduction, Need of operator overloading, rules for operator overloading, overloading the unary and binary operators using member function, operator overloading using friend function, overloading relational and logical operators, overloading new, delete and assignment operator, type conversions
Inheritance: Introduction, Need of inheritance, base and derived classes, member access control, types of inheritance, derived class constructor, constructors in multiple inheritance, overriding member functions, ambiguity in multiple inheritance, virtual base class

UNIT – V Virtual Functions and Templates 7 Hours
Virtual functions: Pointers to objects, ‘this’ pointer, Pointers to derived class, virtual function, rules for virtual function, pure virtual function, abstract class, virtual destructors, early and late binding, container classes
Templates: Introduction, Function template and class template, overloading function template, member function templates and template arguments, Introduction to Standard Template Library (STL), containers, iterators and algorithms

UNIT - VI Exception Handling and File I/O 7 Hours
Namespaces: Introduction, Rules of namespaces
Exception Handling: Introduction, Exception handling mechanism: try, catch and throw, Multiple Exceptions, Exceptions with arguments
Managing Console I/O Operations: Introduction, C++ streams, stream classes, unformatted I/O, formatted I/O and I/O manipulators
File I/O: Introduction, Classes for file stream operations, file operations (open, close, read, write, detect end of file), file modes, File pointers and their manipulations, error handling during file operations

Text Books

Reference Books
214446 : DIGITAL LABORATORY

Teaching Scheme:  
Practical : 2 Hours/Week  
Credits: 01

Examination Scheme:  
Term Work : 25 Marks  
Practical : 50 Marks

Prerequisites: Basic Electronics Engineering

Course Objectives:
1. To learn and understand basic digital design techniques.
2. To learn and understand design and construction of combinational and sequential circuits.
3. To introduce digital logic design software such as VHDL Programming.

Course Outcomes:
After completion of this course student will be able to
1. Spectacle an awareness and apply knowledge and concepts and methods of digital system design techniques as hands-on experiments with the use of necessary A.C, D.C Loading characteristics.
2. Use logic function representation for simplification with K-Maps and analyze as well as design Combinational logic circuits using SSI & MSI chips.
3. Analyze Sequential circuits like Flip-Flops (Truth Table, Excitation table) & design the applications like Asynchronous and Synchronous Counters.
5. Understand the need of skills, techniques and learn state-of-the-art engineering tools through hands-on experimentation on the Xilinx tools for design as well as the basics of VHDL.
6. Understand and implement the design Steps, main programming technique with different modeling styles for any digital circuits with VHDL Programming.

Guidelines for Instructor’s Manual
The faculty member should prepare the laboratory manual for all the experiments and it should be made available to students and laboratory instructor/Assistant. The instructor’s manual should include prologue, university syllabus, conduction & Assessment guidelines, topics under consideration-concept, objectives, outcomes, data sheets of various ICs, 8051 simulator and references.

Guidelines for Student's Lab Journal
1) The laboratory assignments are to be submitted by student in the form of journal. The Journal consists of prologue, Certificate, table of contents, and handwritten write-up of each assignment (Title, Objectives, Problem Statement, Outcomes, software & Hardware requirements, Date of Completion, Assessment grade/marks and assessor's sign, Theory-Concept, circuit diagram, pin configuration, conclusion/analysis).
2) Practical Examination will be based on the term work
3) Candidate is expected to know the theory involved in the experiment
4) The practical examination should be conducted if the journal of the student is completed in all respects and certified by concerned faculty and head of the department
5) All the assignment mentioned in the syllabus must be conducted

Guidelines for Lab/TW Assessment
1) Examiners will assess the term work based on performance of students considering the parameters such as timely conduction of practical assignment, methodology adopted for
implementation of practical assignment, timely submission of assignment in the form of handwritten write-up along with results of implemented assignment, attendance etc.

2) Examiners will judge the understanding of the practical performed in the examination by asking some questions related to theory & implementation of experiments he/she has carried out

3) Appropriate knowledge of usage of necessary simulation software and hardware such as ICs, Registers, digital trainer kits, IC tester should be checked by the faculty member

**Guidelines for Laboratory Conduction**

The instructor is expected to frame the assignments by understanding the prerequisites, technological aspects, utility and recent trends related to the topic. The instructor may set multiple sets of assignments and distribute among batches of students. It is appreciated if the assignments are based on real world problems/applications. Use of open source software is encouraged

The guidelines published by BOS time to time regarding conduction of laboratory assignments and Practical/Oral examination is mandatory.

**Guidelines for Practical Examination**

Both internal and external examiners should jointly set problem statements for practical examination. During practical assessment, the expert evaluator should give the maximum weightage to the satisfactory implementation of the problem statement. The supplementary and relevant questions may be asked at the time of evaluation to judge the student’s understanding of the fundamentals, effective and efficient implementation. The evaluation should be done by both external and internal examiners.

**Suggested List of Laboratory Assignments**

**Group A**

**Combinational Logic Design**

1. Design (truth table, K-map) and implementation of 4-bit BCD to Excess-3 and Excess-3 to BCD Code converters.
2. Design (truth table, K-map) and implementation of 4 bit BCD & Excess 3 Adder using IC7483.
3. Implementation of logic functions using multiplexer IC 74153 & decoder IC 74138. (Verification, cascading & logic function implementation)

**Group B**

**Sequential Logic Design**

1. Design (State diagram, state table & K map) and implementation of 3 bit Up and Down Asynchronous and Synchronous Counter using master slave JK flip-flop IC 7476
2. Design and implementation of Module ‘n’ counter with IC7490 and IC 74191.
3. Design (State Diagram, State Table, K Map) and implementation of Sequence Generator using Shift Register IC 74194.

**Group C**

**VHDL Programming**

Simulation of
1. 4:1 multiplexer using data flow & structural modeling.
2. Full adder using behavioral & structural modeling.
3. 3 bit controlled up / down synchronous counter with preset & clear
Group D

Design, construct digital logic circuits and analyze their behavior through simulation of any one assignment from either Group A or Group B with simulation software like Digital Works 3.0

Student should submit term work in the form of a journal based on the above assignments (Group A, B, and C). Practical examination will be based on the term work. Questions will be asked during the examination to judge the understanding of the practical performed in the examination. Candidate is expected to know the theory involved in the experiment.

Note - Instructor should take care that datasheets of all the required ICs are available in the laboratory & students are verifying the functionality of ICs being used.

Reference Books
214447 : PROGRAMMING LABORATORY

Teaching Scheme:                          Credits                          Examination Scheme:
Practical: 4 Hours/Week                  02                              Term Work: 25 Marks

Prerequisites:                           Practical: 50 Marks
1. Fundamentals of programming languages

Course Objectives:
1. To learn C language constructs and pointers in depth.
2. To learn algorithm development and analysis of algorithms.
3. To learn linear data structures and their applications
4. To learn different searching and sorting techniques

Course Outcomes:
1. Student will be able to apply appropriate constructs of C language, coding standards for application development.
2. Students will be to use dynamic memory allocation concepts and file handling in various application developments.
3. Students will be able to perform basic analysis of algorithms with respect to time and space complexity
4. Students will be able to select appropriate searching and/or sorting techniques in the application development
5. Students will be able to select and use appropriate data structures for problem solving and programming
6. Students will be able to use algorithmic foundations for solving problems and programming

Guidelines for Instructor’s Manual
The faculty member should prepare the laboratory manual for all the experiments and it should be made available to students and laboratory instructor/Assistant. The instructor’s manual should include prologue, university syllabus, conduction & Assessment guidelines, topics under consideration-concept, objectives, outcomes, algorithm written in pseudo language, sample test cases and references.

Guidelines for Student’s Lab Journal
1) The laboratory assignments are to be submitted by student in the form of journal. The Journal consists of prologue, Certificate, table of contents, and handwritten write-up of each assignment (Title, Objectives, Problem Statement, Outcomes, software & Hardware requirements, Date of Completion, Assessment grade/marks and assessor’s sign, Theory-Concept, circuit diagram, pin configuration, conclusion/analysis), printouts of the code written using coding standards, sample test cases etc.
2) Practical Examination will be based on the term work
3) Candidate is expected to know the theory involved in the experiment
4) The practical examination should be conducted if the journal of the candidate is completed in all respects and certified by concerned faculty and head of the department
Guidelines for Lab/TW Assessment

1) Examiners will assess the term work based on performance of students considering the parameters such as timely conduction of practical assignment, methodology adopted for implementation of practical assignment, timely submission of assignment in the form of handwritten write-up along with results of implemented assignment, attendance etc.

2) Examiners will judge the understanding of the practical performed in the examination by asking some questions related to theory & implementation of experiments he/she has carried out

3) Appropriate knowledge of usage of software and hardware such as compiler, debugger, coding standards, algorithm to be implemented etc. should be checked by the concerned faculty member(s)

Guidelines for Laboratory Conduction

1) The instructor is expected to frame the assignments by understanding the prerequisites, technological aspects, utility and recent trends related to the topic. The instructor may set multiple sets of assignments and distribute among batches of students. It is appreciated if the assignments are based on real world problems/applications.

2) The guidelines published by BOS time to time regarding conduction of laboratory assignments and Practical/Oral examination is mandatory.

3) All the assignments should be conducted on multicore hardware and 64-bit open-source software

Guidelines for Practical Examination

Both internal and external examiners should jointly set problem statements for practical examination. During practical assessment, the expert evaluator should give the maximum weightage to the satisfactory implementation of the problem statement. The supplementary and relevant questions may be asked at the time of evaluation to judge the student’s understanding of the fundamentals, effective and efficient implementation. The evaluation should be done by both external and internal examiners.

Suggested List of Laboratory Assignments

1. Represent sets using one dimensional arrays and implement functions to perform
   i. Union
   ii. Intersection
   iii. Difference
   iv. Symmetric difference of two sets

2. Represent matrix using two dimensional arrays and perform following operations with and without pointers:
   i. Addition
   ii. multiplication
   iii. transpose
   iv. Saddle point

3. Implement following operations on string with / without pointers (without using library functions)
i. Length
ii. Palindrome
iii. String comparison
iv. Copy
v. Reverse
vi. Substring

4. Create a Database using array of structures and perform following operations on it:
   i. Create Database
   ii. Display Database
   iii. Add record
   iv. Search record
   v. Modify record
   vi. Delete record

5. a) Sort the set of strings in ascending order using Bubble sort and descending order by using Selection sort or Insertion sort. (Display pass by pass output)
   b) Search a particular string using binary search with and without recursion.

6. Implement sequential file and perform following operations:
   i. Display
   ii. Add records
   iii. Search record
   iv. Modify record
   v. Delete record

7. Implement Quick Sort / Merge Sort to sort the given list of numbers. Display corresponding list in each pass. (with and without recursion)

8. Accept conventional matrix and convert it into sparse matrix using structure and perform addition, simple and fast transpose

9. Implement a singly linked list with following options
   i. Insertion of a node at any location
   ii. Deletion of a node from any location
   iii. display a list
   iv. Display in reverse
   v. Revert the list without using additional data structure.

10. Implement polynomial using CLL and perform
    i. Addition of Polynomials
    ii. Multiplication of polynomials and
    iii. Evaluation of polynomial

11. Implement any database using doubly linked list with following options
    i. Insert a record
    ii. delete a record
    iii. modify a record
    iv. Display list forward
    v. Display list backward

12. Implement Generalized Linked List to create and display the book index.
Note:
1. For all programs implementations students are expected to use meaningful identifiers, proper indentation, use of functions, minimal use of global variables and writing time complexity using any one notation is mandatory.
2. Student should submit term work in the form of a journal based on the above assignments.
3. Practical examination will be based on the term work.
4. Questions will be asked during the examination to judge the understanding of the practical performed in the examination.
5. Candidate is expected to know the theory involved in the experiment.
6. Students are expected to implement at least 3 test cases for each assignment.

Note: This list of assignments is indicative. Concerned faculty member may frame different assignments if required maintaining similar difficulty level.

Reference Books
Teaching Scheme: Practical :2 Hours/Week
Credits: 01
Examination Scheme: Term Work : 25 Marks
Practical : 50 Marks

Prerequisites: Principles of Programming Languages, Fundamentals of Data Structures

Course Objectives:
1. Employ a problem-solving strategy to breakdown a complex problem into a series of simpler tasks.
2. Execute problem-solving actions appropriate to completing a variety of sub problems.
3. Apply analytical and logical thinking to extract facts from a problem description and determine how they relate to one another and to the problems to be solved.
4. Design and implement an object oriented solution to solve a real life problem.
5. Develop problem-solving and programming skills using OOP concept.

Course Outcomes:
After studying this subject student should be able to
1. Break a problem into logical pieces and develop algorithms for solving simple problems.
2. Abstract data and entities from the problem domain, build object models and design software solutions using object-oriented principles and strategies.
4. Develop programs that appropriately utilize key object-oriented concepts.

Guidelines for Instructor’s Manual
The faculty member should prepare the laboratory manual for all the experiments and it should be made available to students and laboratory instructor/Assistant. The instructor’s manual should include prologue, university syllabus, conduction & Assessment guidelines, topics under consideration-concept, objectives, outcomes, algorithm written in pseudo language, sample test cases and references.

Guidelines for Student’s Lab Journal
1) The laboratory assignments are to be submitted by student in the form of journal. The Journal consists of prologue, Certificate, table of contents, and handwritten write-up of each assignment (Title, Objectives, Problem Statement, Outcomes, software & Hardware requirements, Date of Completion, Assessment grade/marks and assessor's sign, Theory-Concept, circuit diagram, pin configuration, conclusion/analysis), printouts of the code written using coding standards, sample test cases etc.
2) Practical Examination will be based on the term work submitted by the student in the form of journal
3) Candidate is expected to know the theory involved in the experiment
4) The practical examination should be conducted if the journal of the candidate is completed in all respects and certified by concerned faculty and head of the department
5) All the assignment mentioned in the syllabus must be conducted

Guidelines for Lab /TW Assessment
1) Examiners will assess the term work based on performance of students considering the parameters such as timely conduction of practical assignment, methodology adopted for
implementation of practical assignment, timely submission of assignment in the form of handwritten write-up along with results of implemented assignment, attendance etc.

2) Examiners will judge the understanding of the practical performed in the examination by asking some questions related to theory & implementation of experiments he/she has carried out.

3) Appropriate knowledge of usage of software and hardware such as compiler, debugger, coding standards, algorithm to be implemented etc. should be checked by the concerned faculty member(s).

Guidelines for Laboratory Conduction

1) The instructor is expected to frame the assignments by understanding the prerequisites, technological aspects, utility and recent trends related to the topic. The instructor may set multiple sets of assignments and distribute among batches of students. It is appreciated if the assignments are based on real world problems/applications.

2) The guidelines published by BOS time to time regarding conduction of laboratory assignments and Practical/Oral examination is mandatory.

3) All the assignments should be conducted on multicore hardware and 64-bit open-source software.

Guidelines for Practical Examination

Both internal and external examiners should jointly set problem statements for practical examination. During practical assessment, the expert evaluator should give the maximum weightage to the satisfactory implementation of the problem statement. The supplementary and relevant questions may be asked at the time of evaluation to judge the student’s understanding of the fundamentals, effective and efficient implementation. The evaluation should be done by both external and internal examiners.

Suggested List of Laboratory Assignments

1. Create a class named weather report that holds a daily weather report with data members day_of_month, hightemp, lowtemp, a mount_rain and amount_snow. Use different types of constructors to initialize the objects. Also include a function that prompts the user and sets values for each field so that you can override the default values. Write a menu driven program in C++ with options to enter data and generate monthly report that displays average of each attribute.

2. A Book shop maintains the inventory of books that are being sold at the shop. The list includes details such as title, author, publisher, price and available stock. Write a program in C++ which will have a class called books with suitable member functions for
   i. Add
   ii. Update
   iii. Search a book
   iv. Purchase a book (update the stock and display the total cost)
   v. Record number of successful/unsuccessful transactions (use static data members to keep count of transactions)

   Use new operator in constructors to allocate memory space required.

3. Design a class ‘Complex ‘with data members for real and imaginary part. Provide default and parameterized constructors. Write a program to perform arithmetic operations of two complex numbers using operator overloading.
i. Addition and subtraction using friend functions
ii. Multiplication and division using member functions

4. Design a base class with name, date of birth, blood group and another base class consisting of the data members such as height and weight. Design one more base class consisting of the insurance policy number and contact address. The derived class contains the data members’ telephone numbers and driving license number.

Write a menu driven program to carry out the following things:
   i. Build a master table   ii. Display  iii. Insert a new entry
   iv. Delete entry  v. Edit  vi. Search for a record

5. Create a base class shape with two double type values and member functions to input the data and compute_area() for calculating area of figure. Derive two classes’ triangle and rectangle. Make compute_area() as a virtual function and redefine this function in the derived class to suit their requirements.

Write a program that accepts dimensions of triangle/rectangle and display calculated area.

6. Write a program in C++ which includes the code for following operations:
   i. A function to read two double type numbers from keyboard
   ii. A function to calculate the division of these two numbers
   iii. A try block to detect and throw an exception if the condition “divide-by-zero” occurs
   iv. Appropriate catch block to handle the exceptions thrown
   v.

7. Write a program in C++ using function/class template to read two matrices of different data types such as integers and floating point values and perform simple arithmetic operations on these matrices separately and display it.

8. Write a program in C++ to implement sequential file for students' database and perform following operations on it
   i) Create Database
   ii) Display Database
   iii) Add a record
   iv) Delete a record
   v) Modify a record

9. Create employee bio-data using following classes
   i) Personal record
   ii) Professional record
   iii) Academic record
   Assume appropriate data members and member function to accept required data & print bio-data. Create bio-data using multiple inheritance using C++.

10. Write a C++ program that creates an output file, writes information to it, closes the file and open it again as an input file and read the information from the file.

Note:
While performing the assignments following care should be taken
1. Proper indenting, coding styles, commenting, naming conventions should be followed.
2. Avoid using global variables as far as possible
3. Faculty should prepare a lab manual including standard test cases & should be available for reference to students.
4. Student should submit term work in the form of a journal based on the above assignments.
5. Practical examination will be based on the term work. Questions will be asked during the
examination to judge the understanding of the practical performed at the time of examination.
6. Candidate is expected to know the theory involved in the experiment.

Note: This list of assignments is indicative. Concerned faculty member may frame different assignments if required maintaining similar difficulty level.

Reference Books
214449 : COMMUNICATION SKILLS

**Teaching Scheme:**
Practical: 2 Hours/Week

**Credits:**
01

**Examination Scheme:**
Term Work: 25 Marks

**Prerequisites:** Basic knowledge of English Language

**Course Objectives:**
1. Improve students’ overall linguistic & communicative competence in English
2. Enhance their pronunciation, vocabulary and LSRW skills
3. Foster their confidence in public speaking and group communication skills

**Course Outcomes:**
1. Provides an ability to understand, analyze and interpret the essentiality of grammar and its proper usage.
2. Build the students’ vocabulary by means of communication via web, direct Communication and indirect communication.
3. Improves Students’ Pronunciation skills and understanding between various phonetic sounds during communication.
4. Understanding the various rules and means of written communication.
5. Effective communication with active listening, facing problems while communication and how to overcome it.

**Course Contents**

**Overview**
The course has been designed for the students of second year Information Technology for enhancing their linguistic and communicative competence. It attempts to give them exposure to the essential linguistic and communication skills by focusing upon the key areas of immediate significance. Students will also be given a theoretical knowledge through lectures about the fundamental concepts in the English language & communication such as grammar, vocabulary, pronunciation and LSRW skills. At the same time adequate practical exposure to these skills will be provided through laboratory sessions. The course aims at striking a fine balance between theory and practice to ensure the all-round improvement of students in these skills. Students will be able to improve their command over communicative English which will enable them to enhance their academic performance and will contribute to their growth as engineering professionals.

**Teaching Methodology in the Language Laboratory**
1. Direct Method – Use of English for communication between the teacher and students. Teachers must emphasize on the use of English in the lab. All the instructions and Interactions must be given in English.
2. Theory lectures should also be interactive and the teacher should encourage students’ participation in the classroom sessions.
3. Laboratory sessions should be activity based and should be conducted in groups and pairs. Guidelines for conducting laboratory sessions have been given below each activity.

**Unit I: ESSENTIAL GRAMMAR AND PHONETICS (5 hrs)**
Tenses: Basic forms and use, sentence formation (general & Technical), Common errors, Parts of speech through context, Direct and reported speech structures and voices, stress & intonation, voice modulation, exercises on pronunciation, use of software for exercises on pronunciation.

Activities:-

- The class of students will always have some students with adequate knowledge of basic grammar and rest with no/poor knowledge.
- The basic grammar exercises can be taught by giving students sentences in their mother tongue and telling them to convert it to English thereby covering parts of speech, tenses, voices, etc.
- The students with acceptable understanding of grammar can be engaged in some advanced grammar exercises like the ones in ‘word power made easy’ or any online exercises mentioned in the references below.
- For intonation, voice modulation, videos by decent orators/movie clips can be shown to the students.
- For pronunciation, exercises based on Homonyms, homophones can be conducted.

Unit II: VOCABULARY ENRICHMENT (5 hrs)
Exposure to words from General Service List (GSL) by West, Academic word list (AWL) by Averil Coxhead (2000) and specific technical terms related to the field of Information technology. Phrases, idioms, proverbs, significant abbreviations, formal (business) vocabulary.

Activities:-

- Students should be given 10 idioms, proverbs and phrases each and should be told to form story using them.
- Students can be divided into teams. Each team should be told to find out 10 new words/phrases the meanings of which should be discussed in the lab. This exercise can be repeated in the last 10 minutes of each lab session so as to add to the students’ vocabulary.

Unit III: WRITING SKILLS

Activities:- students should be made to write letters in formal and informal way like letters, resume, technical report writing.

Unit IV: LISTENING SKILLS (5 hrs)
Types of listening, Levels of Listening, Listening Barriers, Listening Ethics, activities to strengthen students’ listening skills.
Activity:-Chinese whisper
Audio activity:-students should listen to any audio and try to answer question based on that audio.

Unit V: READING SKILLS
Definition, need for reading Skills, techniques for reading, how to develop fluency in Reading.
Lab Activities:
Students can be given some text to read and answer questions related to that text.
Students can be made to read a passage aloud and others can be asked questions based on the passage read.

Unit VI: SPEAKING SKILLS
Difference between talking and Speaking, Attributes /characteristics of public speaking, barriers to effective speaking, Types of speaking: Technical and Non-Technical speaking.
Activities:

- **Prepared speech** (topics are given in advance, students get 10 minutes to prepare the speech and 5 minutes to deliver.
- **Extempore speech** (students deliver speeches spontaneously for 5 minutes each on a given topic)
- **Story telling** (Each student narrates a fictional or real life story for 5 minutes each)
- **Oral review** (Each student orally presents a review on a story or a book read by them)

2. **Power-point Presentations**

Students should make a presentation on any informative topic of their choice. The topic may be technical or non-technical

3. **Formal Group Discussion**

Each batch is divided into two groups of 12 to 14 students each. Two rounds of a GD for each group should be conducted and teacher should give them feedback.

4. **Mock Meetings**

In order to enhance students’ formal oral communication, mock meetings can be conducted. Teacher should give a topic for the meeting and teach students how a notice and agenda for a meeting is prepared. Students will participate in the meeting assuming the roles assigned by the teacher. After the meeting, teacher should guide students on how minutes of meeting are recorded.

6. **Reading and Listening skills**

The batch can be divided into pairs. Each pair will be given an article (any topic) by the teacher. Each pair would come on the stage and read aloud the article one by one. After reading by each pair, the other students will be asked questions on the article by the readers. Students will get marks for correct answers and also for their reading skills. This will evaluate their reading and listening skills. The teacher should give them guidelines on improving their reading and listening skills.

7. **Pronunciation through software or web-based applications**

Teachers should make use of software and web-based applications for giving exercises on pronunciation to students.

8. **Vocabulary exercises through web-based applications**

Teachers should make use of software and web-based applications for giving exercises on vocabulary to students.

9. **Letter, Report & review writing**

Each student will write one formal letter, one report and a review on the topics given by the teacher.

10. **Grammar exercises through web-based applications**

Teachers should make use of software and web-based applications for giving exercises on grammar to students. The term work shall consist of 10 activities carrying 10 marks each. The total marks earned by the students out of 100 will be scaled down to 50. The online exam and term work marks will be further scaled down to 50. Students will have to submit journals or files containing record of each activity performed in laboratory, at the term end.

**References**

1. Rutherford A. J. : Communication skills for Technical Communication, Pearson Education
4. M.S. Rao: Strategies for improving your business communication, SPD
5. Murphy: Essential English Grammar, Cambridge
6. Duttet.al. : A course in Communication Skills, Foundation
7. Patnaik: Group Discussion and Interview Skills, Foundation
9. Lynch: listening, Cambridge
10. Malcom Goodale: Professional Presentations, Cambridge
12. Idioms and proverbs are fun, by Wilco books (author)

ESL Sites (Web-based applications) for vocabulary learning
5. www.englishvocabularyexercises.co

Guidelines for Student’s Lab Journal
1) Student should submit term work in the form of journal which should include handwritten
Write-up, printouts of the code written using coding standards, sample test cases, etc.
2) Practical Examination will be based on the term work
3) Candidate is expected to know the theory involved in the experiment
4) The practical examination should be conducted if journal of the candidate is completed in all
respects and certified by concerned faculty and head of the department
5) All the assignment mentioned in the syllabus must be conducted

Guidelines for Lab /TW Assessment
1) Examiners will assess the term work based on performance of students considering the
parameters such as timely conduction of practical assignment, methodology adopted for
implementation of practical assignment, timely submission of assignment in the form of
handwritten write-up along with results of implemented assignment, attendance etc.
2) Examiners will judge the understanding of the practical performed in the examination by
asking some questions related to theory & implementation of experiments he/she has carried
out
3) Appropriate knowledge of usage of software and hardware such as compiler, debugger, coding
standards, algorithm to be implemented etc. should be checked by the concerned faculty
member(s)

Guidelines for Laboratory Conduction
1) The instructor is expected to frame the assignments by understanding the prerequisites,
technological aspects, utility and recent trends related to the topic. The instructor may set multiple
sets of assignments and distribute among batches of students. It is appreciated if the assignments are
based on real world problems/applications.
2) The guidelines published by BOS time to time regarding conduction of laboratory assignments and
Practical/Oral examination is mandatory.
Audit Course1

In addition to credits course, it is recommended that there should be audit course (non-credit course) preferably in each semester from second year. The student will be awarded grade as AP on successful completion of audit course. The student may opt for one of the audit courses per semester, starting in second year first semester. Though not mandatory, such audit courses can help the student to get awareness of different issues which make impact on human lives and enhance their skill sets to improve their employability. List of audit courses offered in each semester is provided in curriculum. Each student has to choose one audit course from the list per semester. Evaluation of audit course will be done at institute level. Method of conduction and method of assessment for audit courses is suggested.

The student registered for audit course shall be awarded the grade AP and shall be included such grade in the Semester grade report for that course, provided student has the minimum attendance as prescribed by the Savitribai Phule Pune University and satisfactory in-semester performance and secured a passing grade in that audit course. No grade points are associated with this 'AP' grade and performance in these courses is not accounted in the calculation of the performance indices SGPA and CGPA. Evaluation of audit course will be done at institute level itself.


Guidelines for Conduction and Assessment (Any one or more of following but not limited to)

- Lectures/ Guest Lectures
- Visits (Social/Field) and reports
- Demonstrations
- Surveys
- Mini Project
- Hands on experience on specific focused topic

Guidelines for Assessment (Any one or more of following but not limited to)

- Written Test
- Demonstrations/ Practical Test
- Presentations
- IPR/Publication
- Report

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Audit Course Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>210250:AC1-I</td>
<td>Road Safety</td>
</tr>
<tr>
<td>210250:AC1-II</td>
<td>Humanities and Social Sciences</td>
</tr>
<tr>
<td>210250:AC1-III</td>
<td>Environmental Studies</td>
</tr>
<tr>
<td>210250:AC1-IV</td>
<td>Smart Cities</td>
</tr>
</tbody>
</table>

The detail course contents of above mentioned audit courses are available in Computer Engineering 2015 course syllabus.

Moreover students can opt for any other audit course from the list of Audit Course1 of any branch of engineering.
207003 : ENGINEERING MATHEMATICS – III (Common to Computer Engineering)

Teaching Scheme:                                    Credits:                                    Examination Scheme:
Lectures: 4 Hours/Week                               05                                             In-Semester (Online): 50 Marks
Tutorial: 1 Hour/Week                                End-Semester: 50 Marks

Prerequisites:
Differential and Integral Calculus, Taylor series and Infinite series, Differential equations of first order and first degree, Fourier series, Measures of Central tendency and dispersion, Vector algebra, Algebra of complex numbers.

Course Objectives:
After completing this course, students will have adequate mathematical background, conceptual clarity, computational skills and algorithm design for problem solving related to:
1. Linear differential equations of higher order applicable to Control systems, Computer vision, and Robotics.
2. Transform techniques such as Fourier transform, Z-transform and applications to Image processing.
3. Statistical methods such as correlation, regression analysis and probability theory to analyze data and to make predictions applicable to machine intelligence.
4. Vector calculus necessary to analyze and design complex electrical and electronic devices as appropriate to Computer engineering.
5. Complex functions, conformal mappings and contour integration applicable to Image processing, Digital filters and Computer graphics.

Course Outcomes:
At the end of this course, students will be able to:
1. Solve higher order linear differential equation using appropriate techniques for modeling and analyzing electrical circuits.
2. Solve problems related to Fourier transform, Z-Transform and applications to Signal and Image processing.
3. Apply statistical methods like correlation, regression analysis and probability theory for analysis and prediction of a given data as applied to machine intelligence.
4. Perform vector differentiation and integration to analyze the vector fields and apply to compute line, surface and volume integrals.
5. Analyze conformal mappings, transformations and perform contour integration of complex functions required in Image processing, Digital filters and Computer graphics.

Course Contents

UNIT – I  Linear Differential Equations (LDE) and Applications  6 Hours
LDE of n^{th} order with constant coefficients, Method of variation of parameters, Cauchy’s & Legendre’s DE, Simultaneous & Symmetric simultaneous DE. Modeling of Electrical circuits.

UNIT – II  Transforms  6 Hours

UNIT – III Statistics 6 Hours
Measures of central tendency, Standard deviation, Coefficient of variation, Moments, Skewness and Kurtosis, Curve fitting: fitting of straight line, parabola and related curves, Correlation and Regression, Reliability of Regression Estimates.

UNIT – IV Probability and Probability Distributions 6 Hours
Probability, Theorems on Probability, Bayes Theorem, Random variables, Mathematical Expectation, Probability density function, Probability distributions: Binomial, Poisson, Normal and Hypergeometric, Test of Hypothesis: Chi-Square test, t-distribution.

UNIT – V Vector Calculus 6 Hours
Vector differentiation, Gradient, Divergence and Curl, Directional derivative, Solenoidal and Irrotational fields, Vector identities. Line, Surface and Volume integrals, Green’s Lemma, Gauss’s Divergence theorem and Stoke’s theorem.

UNIT - VI Complex Variables 6 Hours
Functions of Complex variables, Analytic functions, Cauchy-Riemann equations, Conformal mapping, Bilinear transformation, Cauchy’s integral theorem, Cauchy’s integral formula, Laurent’s series, and Residue theorem.

Text Books
1. Advanced Engineering Mathematics, 9e, by Erwin Kreyszig (Wiley India).

Reference Books
214450 : COMPUTER GRAPHICS

Teaching Scheme: 
Lectures: 3 Hours/Week

Credits

Examination Scheme: 
In-Semester (Online): 50 Marks
End-Semester: 50 Marks

Prerequisites:
1. Basic Geometry, Trigonometry, Vectors and Matrices
2. Basics of Data Structures and Algorithms

Course Objectives:
1. To acquaint the learners with the basic concepts of Computer Graphics
2. To learn the various algorithms for generating and rendering graphical figures
3. To get familiar with mathematics behind the graphical transformations
4. To understand and apply various methods and techniques regarding projections, animation, shading, illumination and lighting

Course Outcomes:
On completion of the course, learner will be able to –
1. Apply mathematics and logic to develop Computer programs for elementary graphic operations
2. Develop scientific and strategic approach to solve complex problems in the domain of Computer Graphics
3. Develop the competency to understand the concepts related to Computer Vision and Virtual reality
4. Apply the logic to develop animation and gaming programs

Course Contents

UNIT – I BASIC CONCEPTS
Introduction to Computer Graphics, Basics of graphics systems, Raster scan & Random scan displays, basic display processor

Display Files: display file structure, algorithms and display file interpreter. Primitive operations on display file

Plotting Primitives: Scan conversions, lines, line segments, vectors, pixels and frame buffers, vector generation

Line drawing Algorithms: DDA, Bresenham
Circle drawing Algorithms: - DDA, Bresenham
Character Generation: Stroke Principle, Starburst Principle, Bit map method
Introduction to aliasing and anti-aliasing

UNIT – II POLYGONS AND GRAPHICAL TRANSFORMATIONS
Polygon and its types, inside test, polygon filling methods: Seed fill, Scan Line, Flood fill and Boundary fill

2D Geometric Transformations - translation, scaling, rotation, other transformations such as reflection, shearing, matrix representation and homogeneous coordinate system, Composite transformations

UNIT – III 3D TRANSFORMATIONS AND PROJECTIONS

8 Hours

6 Hours

6 Hours
Translation, scaling, rotation, rotation about X, Y, Z and arbitrary axis reflection about XY, YZ, XZ and arbitrary plane.

**Projections:** Types Parallel - Oblique: Cavalier, Cabinet and orthographic : Isometric, Dimetric, Trimetric and Perspective - Vanishing Points as 1 point, 2 point and 3 point

---

**UNIT – IV  SEGMENTS, WINDOWING AND CLIPPING**  
**6 Hours**

Segment: Introduction, Segment table, Segment creation, closing, deleting and renaming, Visibility

Windowing: Concept of window and viewport, viewing transformations

Line Clipping: Cohen Sutherland Method, Midpoint subdivision method

Polygon Clipping: Sutherland Hodgman method for clipping convex and concave polygons

---

**UNIT – V  SHADING, ANIMATION AND GAMING**  
**6 Hours**

Shading: Halftoning, Gouraud and Phong Shading

Computer Animation: Animation sequences, functions & Languages, Key-frame Systems, Motion Specifications.

Gaming platforms: Graphics Memory Pipeline, Block diagram of NVIDIA workstation and i860

Introduction to OpenGL ES

---

**UNIT - VI  CURVES AND FRACTALS**  
**6 Hours**

Introduction, Curve generation, Interpolation, interpolating algorithms, interpolating polygons, B-Splines and corners, Bezier curves, Fractals, fractal lines and surfaces

Interactive Graphics & usage of the tools of computer graphics – 3D Studio and Maya

---

**Text Books**


**Reference Books**

Prerequisites: Computer Organization & Architecture

Course Objectives:
1. To study architecture and features of 80386 microprocessors and 8051 microcontroller
2. To learn design of minimum system using 8051 micro-controller.

Course Outcomes:
1. Students will learn architectural details of 80386 microprocessor
2. Students will be able to explain memory management and multitasking of 80386 microprocessor
3. Students will understand architecture and memory organization of 8051 microcontroller
4. Students will be able to explain timers and interrupts of 8051 microcontroller and its interfacing with I/O devices

Course Contents

UNIT – I  INTRODUCTION TO ASSEMBLY LANGUAGE PROGRAMMING & 80386 PROCESSOR 8 Hours
Introduction to assembly language programming, ALP tools- Assembler, Linker, Loader, Debugger, Emulator, Assembler directives, Far and near procedure, Macros, DOS Internals, DOS Calls.
80386 - Features and Architecture, Register Set, 80386 Real mode segmentation and Address translation, Addressing modes, Instruction set.

UNIT – II  80386 MEMORY MANAGEMENT 8 Hours
Pin Description of 80386, 16/32-bit data transfer mechanism, Pipelined & Non pipelined bus cycles. Segmentation - support registers and Data structures, Descriptors, Memory management through segmentation, Logical to linear/physical address translation. Privileged instructions, Protection in segmentation, Inter-privilege level transfer using Call gates and confirming code segment.

UNIT – III  80386 – PRIVILEGE PROTECTION, MULTITASKING & INTERRUPTS, EXCEPTIONS 8 Hours
Paging - support registers and Data structures, Descriptors, Linear to physical address translation, Page level protection. Multitasking - Support registers and Data structures, Descriptors, Task switching. Real and Protected mode Interrupt structure - IVT, IDT, Type of exceptions and Processing.

UNIT – IV  INTRODUCTION TO 8051 MICROCONTROLLER 8 Hours
Difference between microprocessor and microcontroller, 8051 microcontroller - Features, Architecture, Pin Description. On-Chip data memory and program memory organization - Register set, Register bank and Special Function Registers (SFRs). Addressing modes, Instruction set. External data memory and program memory organization.
UNIT – V  PORTS, INTERRUPTS & TIMERS/COUNTERS OF 8051  8 Hours
I/O ports programming - Structures, Related SFRs and Configuration.
Interrupt programming - Structure and Response, Related SFRs and Configuration.
Timers/counters programming - Structure, Related SFRs, Operating modes, Delay calculations and Configuration.
Serial port programming - Related SFRs, Operating modes, Baud rate calculation and Configuration.

UNIT - VI  8051 INTERFACING & APPLICATIONS  8 Hours
PPI 8255 – Features, Architecture, Operating modes & Programming.
Interfacing of displays: LED, LCD, Seven segments.
Keyboard Interfacing, Interfacing of ADC and DAC, Interfacing of stepper motor, Interfacing of Sensors (temperature, pressure), External data memory and program memory interfacing, Design of minimum system using 8051 micro-controller for various applications.

Text Books

Reference Books
3. Peter Abel, NiyazNizamuddin, "IBM PC Assembly Language and Programming", Pearson Education
### 214452 : DATA STRUCTURES AND FILES

**Teaching Scheme:**  
Lectures: 4 Hours/Week  
Credits: 04

**Examination Scheme:**  
In-Semester (Online): 50 Marks  
End-Semester: 50 Marks

**Prerequisites:** Fundamentals of Data Structures, Discrete Structures

**Course Objectives:**
1. To study data structures and their implementations using OOP (C++) and their applications.
2. To study some advanced data structures such as trees, graphs and tables.
3. To learn different file organizations.

**Course Outcomes:**
1. Basic ability to analyze algorithms and to determine algorithm correctness and time efficiency class.
2. Understand different advanced abstract data type (ADT) and data structures and their implementations.
3. Understand different algorithm design techniques (brute-force, divide and conquer, greedy, etc.) and their implementation.
4. Ability to apply and implement learned algorithm design techniques and data structures to solve problems.

**Course Contents**

**UNIT – I STACKS AND QUEUES**  
8 Hours  
Concept of stack, stack as ADT, Implementation of stack using linked organization. Concept of implicit and explicit stack, Applications of stack.  
Concept of queues as ADT, Implementation of queue using linked organization. Concept of circular queue, double ended queue and priority queue. Applications of queues.

**UNIT – II TREES**  
10 Hours  
Difference in linear and non-linear data structure, Trees and binary trees-concept and terminology. Expression tree. Conversion of general tree to binary tree. Binary tree as an ADT. Recursive and non-recursive algorithms for binary tree traversals, Binary search trees, Binary search tree as ADT, Applications of trees

**UNIT – III GRAPHS**  
8 Hours  
Graph as an ADT, Representation of graphs using adjacency matrix and adjacency list, Depth First Search and Breadth First Search traversal. Prim’s and Kruskal’s algorithms for minimum spanning tree, shortest path using Warshall’s and Dijkstra’s algorithm, topological sorting.

**UNIT – IV TABLES**  
8 Hours  
Symbol Table: Notion of Symbol Table, OBST, Huffman's algorithm, Heap data structure, Min and Max Heap, Heap sort implementation, applications of heap  
Hash tables and scattered tables: Basic concepts, hash function, characteristics of good hash function, different key-to-address transformations techniques, synonyms or collisions, collision resolution techniques- linear probing, quadratic probing, rehashing, chaining without replacement and chaining with replacement.
UNIT – V   ADVANCE TREES  
7 Hours  
Concept of threaded binary tree. Preorder and In-order traversals of in-order threaded binary tree, 
Concept of red and black trees, AVL Trees, B trees, B+ trees, Splay trees  

UNIT - VI   FILE ORGANIZATION  
7 Hours  
External storage devices, File, File types and file organization (sequential, index sequential and Direct 
access), Primitive operations and implementations for each type and comparison  

Text Books  
1.  R. Gilberg, B. Forouzan, "Data Structures: A pseudo Code Approach with C++", Cengage 
    Learning, ISBN 9788131503140.  

Reference Books  
1.  Bruno R Preiss, “Data Structures and Algorithms with Object-Oriented Design Patterns in C++”, 
    Wiley India Edition  
5.  J. Tremblay, P. Soresan, "An Introduction to Data Structures with Applications", 2nd edition, 
214453 : FOUNDATIONS OF COMMUNICATION AND COMPUTER NETWORK

Teaching Scheme:  
Lectures: 4 Hours/Week  
Credits: 04

Examination Scheme:  
In-Semester (Online): 50 Marks  
End-Semester: 50 Marks

Prerequisites: Discrete Structures, Engineering Mathematics I and II

Course Objectives:  
1. To understand fundamentals of communication systems  
2. To acquaint themselves with layered model used computer networks

Course Outcomes:  
After successful completion of this course, student will be able to  
1. Understand data/signal transmission over communication media  
2. Recognize usage of various modulation techniques in communication  
3. Analyze various spread spectrum and multiplexing techniques  
4. Use concepts of data communication to solve various related problems  
5. Understand error correction and detection techniques.  
6. Acquaint with transmission media and their standards

Course Contents

UNIT – I  INTRODUCTION TO COMMUNICATION SYSTEMS  6 Hours
Introduction To Communication Theory: Terminologies, Elements Of Analog Communication System, Baseband signal, Band-pass signal, Need For Modulation, Electromagnetic Spectrum And Typical Applications, Basics Of Signal (Analog And Digital,) Representation And Analysis (Time and frequency)
Introduction To Transmission Media: Guided Media : Twisted-Pair Cable, Coaxial Cable And Fiber-Optic Cable, Unguided Media: Wireless , Radio Waves, Microwaves And Infrared
Noise: External Noise, Internal Noise ,Noise Calculations

UNIT – II  AMPLITUDE AND ANGLE MODULATION  6 Hours
Angle Modulation Techniques: Theory Of Angle Modulation Techniques, Practical Issues In Frequency Modulation, Generation Of Frequency Modulation, Frequency Spectrum

UNIT – III  PULSE AND DIGITAL MODULATION TECHNIQUES  6 Hours
Average Information, Entropy, Information Rate. Source coding: Shanon-Fano, Huffman and Limpel-Ziv

UNIT – IV  ERROR CONTROL CODING  6 Hours
Error Detection And Correction:
Introduction, Block Coding (Error Detection, Error Correction, Hamming Distance And Minimum Hamming Distance), Linear Block Codes, Cyclic Codes: CRC (Hardware Implementation, Polynomials), Advantages Of Cyclic Codes, Other Cyclic Codes As Examples: CHECKSUM: One’s Complement, Internet Checksum
Stop-and-Wait Automatic Repeat Request, Go-Back-N Automatic Repeat Request, Selective Repeat Automatic Repeat Request

UNIT – V MULTIPLEXING AND MULTIPLE ACCESS 6 Hours
Multiplexing: FDM, TDM, Synchronous Time-Division Multiplexing, Statistical Time-Division Multiplexing, Spread Spectrum: FHSS And DSSS
Random access: (ALOHA, CSMA, CSMA/CD And CSMA/CA), Controlled Access (Reservation, Polling And Token Passing) Channelization (FDMA, TDMA And CDMA)

UNIT - VI MULTIPLE ACCESS AND PHYSICAL & MAC LAYER STANDARDS 6 Hours
LAN hardware: (Switches, router, hub, bridge and their types)
IEEE 802.3, Fast Ethernet (Mac Sublayer & Physical Layer), Gigabit Ethernet (Mac Sublayer, Physical Layer) Ten-Gigabit Ethernet, Token ring and token bus standard
Circuit Switched Networks, Packet (Datagram) Networks, Virtual Circuits, Structure Of Circuit And Packet Switches

Text Books
2. Data Communications and Networking Behrouz AForouzan. - 4th Ed, Mgh

Reference Books
1. Introduction to Analog and Digital Communications by Simon Haykin and Michael Moher, John Wiley & Sons, Inc.
3. Computer Networks by A S Tanenbaum
4. Data Communications and networking an engineering approach by jamesirvin, wiley
5. electronic communications by Roddy & Coolen, Phi.
6. Electronic Communication System by Kenedy & Davis, TMH
214454 : PROCESSOR INTERFACING LABORATORY

Teaching Scheme:  
Practical : 4 Hours/Week  
Credits: 02

Examination Scheme:  
Term Work: 25 Marks  
Practical: 50 Marks

Prerequisites: Processor Architecture and Interfacing, Computer Organization and Architecture

Course Objectives:
1. To learn assembly language programming of 80386 microprocessors and 8051 microcontrollers.
2. To learn interfacing of real world input and output devices to 8051 microcontroller

Course Outcomes:
1. Students will learn concepts related to assembly language programming
2. Students will be able to write and execute assembly language program to perform array addition, code conversion, block transfer, sorting and string operations
3. Students will be able to learn interfacing of real world input and output devices to 8051 microcontroller

Guidelines for Instructor's Manual
The faculty member should prepare the laboratory manual for all the experiments and it should be made available to students and laboratory instructor/Assistant. The instructor's manual should include prologue, university syllabus, conduction & Assessment guidelines, topics under consideration-concept, objectives, outcomes, algorithm, sample test cases and references etc.

Guidelines for Student's Lab Journal
1) The laboratory assignments are to be submitted by student in the form of journal. The Journal consists of prologue, Certificate, table of contents, and handwritten write-up of each assignment (Title, Objectives, Problem Statement, Outcomes, software & Hardware requirements, Date of Completion, Assessment grade/marks and assessor's sign, Theory-Concept, circuit diagram, pin configuration, conclusion/analysis), printouts of the code written using coding standards, sample test cases etc.
2) Practical Examination will be based on the term work submitted by the student in the form of journal
3) Candidate is expected to know the theory involved in the experiment
4) The practical examination should be conducted if the journal of the candidate is completed in all respects and certified by concerned faculty and head of the department
5) All the assignment mentioned in the syllabus must be conducted

Guidelines for Lab /TW Assessment
1) Examiners will assess the term work based on performance of students considering the parameters such as timely conduction of practical assignment, methodology adopted for implementation of practical assignment, timely submission of assignment in the form of handwritten write-up along with results of implemented assignment, attendance etc.
2) Examiners will judge the understanding of the practical performed in the examination by asking some questions related to theory & implementation of experiments he/she has carried out
3) Necessary knowledge of usage of software and hardware such as assembler, linker, debugger, 8051 microcontrollers and its interfacing kits should be checked by the concerned faculty
Guidelines for Laboratory Conduction
1) The instructor is expected to frame the assignments by understanding the prerequisites, technological aspects, utility and recent trends related to the topic. The instructor may set multiple sets of assignments and distribute among batches of students. It is appreciated if the assignments are based on real world problems/applications.
2) The guidelines published by BOS time to time regarding conduction of laboratory assignments and Practical/Oral examination is mandatory.

Guidelines for Practical Examination
Both internal and external examiners should jointly set problem statements for practical examination. During practical assessment, the expert evaluator should give the maximum weightage to the satisfactory implementation of the problem statement. The supplementary and relevant questions may be asked at the time of evaluation to judge the student’s understanding of the fundamentals, effective and efficient implementation. The evaluation should be done by both external and internal examiners.

Suggested List of Laboratory Assignments

Group A: Microprocessor Programming
1. Write Assembly Language Program (ALP) to add array of N numbers stored in the memory.

2. Write menu driven ALP to convert 4-digit Hex number into its equivalent BCD number and 5-digit BCD number into its equivalent HEX number. Make your program user friendly to accept the choice from user for
   i. HEX to BCD  
   ii. BCD to HEX  
   iii. EXIT.
   Display proper strings to prompt the user while accepting the input and displaying the result. Write near procedures to complete the task.

3. Write ALP to perform following operation on string:
   i. Find and display length
   ii. Display reverse
   iii. Check whether string is palindrome or not.
   Display proper strings to prompt the user while accepting the input and displaying the result. Write near procedures to complete the task.

4. Write menu driven ALP to perform string manipulations. The strings to be accepted from the user is to be stored in code segment Module_1 and write FAR PROCEDURES in code segment Module_2 to perform any two of the following string operations:
   i. Concatenation of two strings.
   ii. Comparison of two strings.
   iii. Finding Number of occurrences of a sub-string in the given string
   iv. Finding number of alphabets, digits, special characters, lower & upper case alphabets, words and number of lines from the text.

   Note: Use PUBLIC and EXTERN directives. Create .OBJ files of both the modules and link them to create an .EXE file.
5. Assignment on file operations
   Select any one of the following assignments
   a. Write menu driven program in C using int86, int86x, intdos and intdosx functions for
      implementing following operations on file.
      i. To delete a file
      ii. To create a directory
      iii. To copy a file
   b. Write 8086 ALP to read command line arguments using Program Segment Prefix (PSP) and
      simulate “DOS COPY Command”. Use file handle function for handling the files. Handle all
      the errors and display appropriate message if user does not enter proper command line
      argument.

Group B: Microcontroller Programming
Assignment 6 and 7. Select any two of the following assignments:
   i. Write 8051 ALP to add n, 8 bits numbers found in internal ram location 40H onwards and
      store results in R6 and R7.
   ii. Write 8051 ALP to multiply 16 bit number by 8 bit number and store the result in internal
      memory location.
   iii. Write 8051 ALP for block transfer for internal / external memory.
   iv. Write 8051 ALP for sorting byte array in ascending / descending order.

8. Select any one of the following assignments.
   i. Timer programming: ISR based
      Write ALP to generate 2 KHz square wave using Timer interrupt on any port pin.
   ii. Serial port programming: ISR based
      Connect two 8051 microcontrollers using serial ports. Send FFh and 00H alternatively to
      receiver. Output received byte to port1, see port1 pin waveform on CRO.

9 & 10. Select any two of the following assignments:
   Write ALP to interface 8051 with:
   i. DAC to generate square, triangular and trapezoidal waveforms.
   ii. ADC to read and display equivalent digital output.
   iii. Stepper motor to rotate motor with different step angles and speeds.
   iv. Sensors (temperature, pressure) to read and display values of the physical parameters
       sensed.
   v. LCD to display message.

Note: This list of assignments is indicative. Concerned faculty member may frame different
assignments if required maintaining similar difficulty level.

Reference Books
1. Peter Abel, NiyazNizamuddin, "IBM PC Assembly Language and Programming", Pearson Education
214455 : DATA STRUCTURE AND FILES LABORATORY

Teaching Scheme: Practical : 4 Hours/Week 02

Examination Scheme:
Term Work : 25 Marks
Practical : 50 Marks

Prerequisites: Fundamentals of Data Structures, Discrete Structures

Course Objectives:
1. To study data structures and their implementations using OOP (C++) and their applications.
2. To study some advanced data structures such as trees, graphs and tables.
3. To learn different file organizations.

Course Outcomes:
1. Basic ability to analyze algorithms and to determine algorithm correctness and time efficiency class.
2. Understand different advanced abstract data type (ADT) and data structures and their implementations.
3. Understand different algorithm design techniques (brute-force, divide and conquer, greedy, etc.) and their implementation
4. Ability to apply and implement learned algorithm design techniques and data structures to solve problems.

Guidelines for Instructor's Manual
The faculty member should prepare the laboratory manual for all the experiments and it should be made available to students and laboratory instructor/Assistant. The instructor's manual should include prologue, university syllabus, conduction & Assessment guidelines, topics under consideration-concept, objectives, outcomes, algorithm written in pseudo language, sample test cases and references.

Guidelines for Student's Lab Journal
1) The laboratory assignments are to be submitted by student in the form of journal. The Journal consists of prologue, Certificate, table of contents, and handwritten write-up of each assignment (Title, Objectives, Problem Statement, Outcomes, software & Hardware requirements, Date of Completion, Assessment grade/marks and assessor's sign, Theory-Concept, circuit diagram, pin configuration, conclusion/analysis), printouts of the code written using coding standards, sample test cases etc.
2) Practical Examination will be based on the term work submitted by the student in the form of journal
3) Candidate is expected to know the theory involved in the experiment
4) The practical examination should be conducted if the journal of the candidate is completed in all respects and certified by concerned faculty and head of the department
5) All the assignment mentioned in the syllabus must be conducted

Guidelines for Lab/TW Assessment
1) Examiners will assess the term work based on performance of students considering the parameters such as timely conduction of practical assignment, methodology adopted for
implementation of practical assignment, timely submission of assignment in the form of handwritten write-up along with results of implemented assignment, attendance etc.

2) Examiners will judge the understanding of the practical performed in the examination by asking some questions related to theory & implementation of experiments he/she has carried out.

3) Appropriate knowledge of usage of software and hardware such as compiler, linker, debugger, coding standards, algorithms to be implemented should be checked by the concerned faculty member(s).

Guidelines for Laboratory Conduction

1) The instructor is expected to frame the assignments by understanding the prerequisites, technological aspects, utility and recent trends related to the topic. The instructor may set multiple sets of assignments and distribute among batches of students. It is appreciated if the assignments are based on real world problems/applications.

2) The guidelines published by BOS time to time regarding conduction of laboratory assignments and Practical/Oral examination is mandatory.

3) All the assignments should be implemented using C++.

4) All the assignments should be conducted on multicore hardware and 64-bit open-source software.

Guidelines for Practical Examination

Both internal and external examiners should jointly set problem statements for practical examination. During practical assessment, the expert evaluator should give the maximum weightage to the satisfactory implementation of the problem statement. The supplementary and relevant questions may be asked at the time of evaluation to judge the student’s understanding of the fundamentals, effective and efficient implementation. The evaluation should be done by both external and internal examiners.

Suggested List of Laboratory Assignments

1. Implement stack as an abstract data type using linked list and use this ADT for conversion of infix expression to postfix, prefix and evaluation of postfix and prefix expression.

2. Implement priority queue as ADT using single linked list for servicing patients in an hospital with priorities as i) Serious (top priority) ii) medium illness (medium priority) iii) General (Least priority).

3. Create Binary tree and perform following operations:
   a. Insert
   b. Display
   c. Depth of a tree
   d. Display leaf-nodes
   e. Create a copy of a tree

4. Construct and expression tree from postfix/prefix expression and perform recursive and non-recursive In-order, pre-order and post-order traversals.

5. Implement binary search tree and perform following operations:
   a. Insert
   b. Delete
   c. Search
   d. Mirror image
   e. Display
6. Consider a friends’ network on face book social web site. Model it as a graph to represent each node as a user and a link to represent the friend relationship between them. Store data such as date of birth, number of comments for each user.
   1. Find who is having maximum friends
   2. Find who has post maximum and minimum comments
   3. Find users having birthday in this month.
   Hint: (Use adjacency list representation and perform DFS and BFS traversals)

7. Represent any real world graph using adjacency list/adjacency matrix find minimum spanning tree using Kruskal’s algorithm.

8. Represent a given graph using adjacency matrix/adjacency list and find the shortest path using Dijkstra’s algorithm (single source all destination).

9. Store data of students with telephone no and name in the structure using hashing function for telephone number and implement chaining with and without replacement.

10. A business house has several offices in different countries; they want to lease phone lines to connect them with each other and the phone company charges different rent to connect different pairs of cities. Business house want to connect all its offices with a minimum total cost. Solve the problem by suggesting appropriate data structures

11. Department maintains a student information. The file contains roll number, name, division and address. Write a program to create a sequential file to store and maintain student data. It should allow the user to add, delete information of student. Display information of particular employee. If record of student does not exist an appropriate message is displayed. If student record is found it should display the student details.

12. Implement direct access file using hashing (chaining without replacement) perform following operations on it
   a. Create Database
   b. Display Database
   c. Add a record
   d. Search a record
   e. Modify a record

Note: This list of assignments is indicative. Concerned faculty member may frame different assignments if required maintaining similar difficulty level.

Reference Books
214456: COMPUTER GRAPHICS LABORATORY

Teaching Scheme: Credits
Practical: 2 Hours/Week 01

Examination Scheme:
Term Work: 25 Marks
Practical: 50 Marks

Prerequisites:
1. Basic Geometry, Trigonometry, Vectors and Matrices
2. Basics of Data Structures and Algorithms

Course Objectives:
1. To acquaint the learners with the basic concepts of Computer Graphics
2. To learn the various algorithms for generating and rendering graphical figures
3. To get familiar with mathematics behind the graphical transformations
4. To understand and apply various methods and techniques regarding projections, animation, shading, illumination and lighting

Course Outcomes:
On completion of the course, learner will be able to –
1. Apply mathematics and logic to develop Computer programs for elementary graphic operations
2. Develop scientific and strategic approach to solve complex problems in the domain of Computer Graphics
3. Develop the competency to understand the concepts related to Computer Vision and Virtual reality
4. Apply the logic to develop animation and gaming programs

Guidelines for Instructor’s Manual
The faculty member should prepare the laboratory manual for all the experiments and it should be made available to students and laboratory instructor/Assistant. The instructor’s manual should include prologue, university syllabus, conduction & Assessment guidelines, topics under consideration-concept, objectives, outcomes, algorithm written in pseudo language, sample test cases and references.

Guidelines for Student’s Lab Journal
1) The laboratory assignments are to be submitted by student in the form of journal. The Journal consists of prologue, Certificate, table of contents, and handwritten write-up of each assignment (Title, Objectives, Problem Statement, Outcomes, software & Hardware requirements, Date of Completion, Assessment grade/marks and assessor’s sign, Theory-Concept, circuit diagram, pin configuration, conclusion/analysis), printouts of the code written using coding standards, sample test cases etc.
2) Practical Examination will be based on the term work submitted by the student in the form of journal
3) Candidate is expected to know the theory involved in the experiment
4) The practical examination should be conducted if the journal of the candidate is completed in all respects and certified by concerned faculty and head of the department
5) All the assignment mentioned in the syllabus must be conducted
Guidelines for Lab /TW Assessment
1) Examiners will assess the term work based on performance of students considering the parameters such as timely conduction of practical assignment, methodology adopted for implementation of practical assignment, timely submission of assignment in the form of handwritten write-up along with results of implemented assignment, attendance etc.
2) Examiners will judge the understanding of the practical performed in the examination by asking some questions related to theory & implementation of experiments he/she has carried out
3) Appropriate knowledge of usage of software and hardware such as compiler, linker, debugger, coding standards, algorithms to be implemented should be checked by the concerned faculty member(s)

Guidelines for Laboratory Conduction
5) The instructor is expected to frame the assignments by understanding the prerequisites, technological aspects, utility and recent trends related to the topic. The instructor may set multiple sets of assignments and distribute among batches of students. It is appreciated if the assignments are based on real world problems/applications.
6) The guidelines published by BOS time to time regarding conduction of laboratory assignments and Practical/Oral examination is mandatory
7) All the assignments should be conducted on multicore hardware and 64-bit open-source software
8) All the assignments should be conducted preferably using OpenGL or Linux platform
9) Implement any 4 assignments from group A and any 4 assignments from group B

Guidelines for Practical Examination
Both internal and external examiners should jointly set problem statements for practical examination. During practical assessment, the expert evaluator should give the maximum weightage to the satisfactory implementation of the problem statement. The supplementary and relevant questions may be asked at the time of evaluation to judge the student’s understanding of the fundamentals, effective and efficient implementation. The evaluation should be done by both external and internal examiners.

Suggested List of Laboratory Assignments

Group A
1. A Mandelbrot Set is a set of complex number z that does not diverge under the transformation 
   \[ z = z^2 + c \] 
   with \( c = 0 \). Where, both \( x \) and \( z \) represent the complex numbers.
   a) Plot the Mandelbrot set for the threshold \(|x| = 2\).
   b) Plot Julia set choosing \( z \neq 0 \). Use 254 colors for plotting in both cases. Change the threshold to observe different patterns.
2. Draw the polygons by using the mouse. Choose colors by clicking on the designed color pane.
   Use window port to draw. (Use DDA algorithm for line drawing)
3. Draw inscribed and Circumscribed circles in the triangle as shown as an example below. (Use any Circle drawing and Line drawing algorithms)
4. Draw the following pattern using any Line drawing algorithms.

5. Draw a 4X4 chessboard rotated 45° with the horizontal axis. Use Bresenham algorithm to draw all the lines. Use seed fill algorithm to fill black squares of the rotated chessboard.

**Group B**

1. Implement Cohen Sutherland Hodgman algorithm to clip any given polygon. Provide the vertices of the polygon to be clipped and pattern of clipping interactively.
2. Implement translation, sheer, rotation and scaling transformations on equilateral triangle and rhombus.
3. Implement Cube rotation about vertical axis passing through its centroid.
4. Generate fractal patterns by using Koch curves.
5. Animation: Implement any one of the following animation assignments,
   i) Clock with pendulum
   ii) National Flag hoisting
   iii) Vehicle/boat locomotion
   iv) Falling Water drop into the water and generated waves after impact
   v) Kaleidoscope views generation (at least 3 colorful patterns)

**Note:** This list of assignments is indicative. Concerned faculty member may frame different assignments if required maintaining similar difficulty level.

**Reference Books**

Audit Course2

In addition to credits course, it is recommended that there should be audit course (non-credit course) preferably in each semester from second year. The student will be awarded grade as AP on successful completion of audit course. The student may opt for one of the audit courses per semester, starting in second year first semester. Though not mandatory, such audit courses can help the student to get awareness of different issues which make impact on human lives and enhance their skill sets to improve their employability. List of audit courses offered in each semester is provided in curriculum. Each student has to choose one audit course from the list per semester. Evaluation of audit course will be done at institute level. Method of conduction and method of assessment for audit courses is suggested.

The student registered for audit course shall be awarded the grade AP and shall be included such grade in the Semester grade report for that course, provided student has the minimum attendance as prescribed by the Savitribai Phule Pune University and satisfactory in-semester performance and secured a passing grade in that audit course. No grade points are associated with this 'AP' grade and performance in these courses is not accounted in the calculation of the performance indices SGPA and CGPA. Evaluation of audit course will be done at institute level itself.


Guidelines for Conduction and Assessment (Any one or more of following but not limited to)

- Lectures/ Guest Lectures
- Visits (Social/Field) and reports
- Demonstrations
- Surveys
- Mini Project
- Hands on experience on specific focused topic

Guidelines for Assessment (Any one or more of following but not limited to)

- Written Test
- Demonstrations/ Practical Test
- Presentations
- IPR/Publication
- Report

List of courses under Audit Course2

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Audit Course Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>210258:AC2-I</td>
<td>Water Management</td>
</tr>
<tr>
<td>210258:AC2-II</td>
<td>Intellectual Property Rights and Patents</td>
</tr>
<tr>
<td>210258:AC2-III</td>
<td>The Science of Happiness</td>
</tr>
<tr>
<td>210258:AC2-IV</td>
<td>Stress Relief: Yoga and Meditation</td>
</tr>
</tbody>
</table>

The detail course contents of above mentioned audit courses are available in Computer Engineering 2015 course.

Moreover students can opt for any other audit course from the list of Audit Course2 of any branch of engineering.