CE 2101 PRINCIPLES OF PROGRAMMING LANGUAGES

Teaching Scheme
Lecture : 3 Hrs/Week

Examination Scheme
In Semester : 25 Marks
End Semester : 50 Marks
Credits : 3

Prerequisite:
ES 1202 Fundamentals of Programming Languages - II

Course Objectives:
To facilitate the learners
1) To understand structural, computational and functional aspects regarding programming languages.
2) To understand and apply object-oriented principles for application development.
3) To develop programming applications using Java.
4) To understand language design concepts for procedural languages.
5) To analyse and evaluate pros and cons of various programming paradigms.

Course Outcome:
By taking this course, the learner will be able to
1) Understand object-oriented concepts with simple examples in C++, Java and python
2) Apply object-oriented principles for effective programming.
3) Develop simple programs using object-oriented programming language Java.
4) Analyse the strength and weakness of Java programming languages for effective and efficient program development.
5) Evaluate different programming paradigms for application development.

Unit 1: Introduction                      (08)

Unit 2: Object-Oriented Programming with Java                         (08)
Java History, Java Features, Java And Internet, Java Virtual Machine, Class, Object, Methods, Constructors, this Keyword. Garbage Collection, Finalize Method, Argument Passing, Function Overloading, Constructor Overloading. Access Control, Static, Final, Arrays, Inheritance, Base Class and Derived Class, Protected Members, Constructor in Derived Class. Concept of Polymorphism, Abstract Classes, Overriding Member Functions, Super Keyword.

Unit 3: Interfaces, Exception Handling and Collections                           (06)
Interfaces, Package, Exception Fundamentals, Try, Catch, Throw, Throws, Finally, Built-In Exceptions, Custom Exceptions. Java Collection Framework Overview, Collection Interfaces, Collection Classes : ArrayList, Accessing Collection via Iterator. Basic Input Output in Java.
Unit 4: Language Design Concepts
Programming Language Design, Programming Language Processing. Data Types: Primitive Data Types, Composite Data Types, Recursive Data Types, Implementation and Storage Representation of Data Types. Type Binding, Binding and Binding Times, Type Checking, Type Conversion, Expressions, Statements.

Unit 5: Procedural Programming

Unit 6: Functional Programming

Text Books:

Reference Books:
CE 2102 DATA STRUCTURES AND ALGORITHMS I

Teaching Scheme
Lectures: 3 Hrs/Week
Tutorials: 1Hr/Week

Examination Scheme
In Semester : 50 Marks
End Semester : 50 Marks
Credits : 4

Prerequisite:
1. ES 1202 Fundamentals of Programming Language - II

Course Objectives:
To facilitate the learners:
1. To recall and understand the concepts of problem solving, algorithms and data structures.
2. To understand data representation, implementation and applications of linear data structures.
3. To analyze algorithms using time and space complexity.
4. To learn, apply and analyze various data searching and sorting techniques.

Course Outcomes:
By taking this course, the learner will be able to:
1. Design an algorithmic solution for given problem.
2. Distinguish between various linear data structures based on their representations and applications.
3. Analyze and compare algorithms using time and space complexity.
4. Classify various data searching and sorting algorithms.

Unit 1: Introduction to Algorithm, Data Structures and Analysis of Algorithms (07)

Unit 2: Linear Data Structures Using Sequential Organization (06)
Concept of Sequential Organization, Concept of Linear Data Structures, Array as an ADT, Storage Representation of an Array – Row major and Column major, Introduction to Multidimensional Arrays. Concept of Ordered List, Application: Polynomial as an ADT using Array. Introduction to Strings and operations on Strings. Concept of Sparse Representation of Sparse matrix, Sparse matrix operations.

Unit 3: Sorting and Searching Techniques (08)

Unit 4: Linked List (08)
Unit 5: Stacks


Unit 6: Queues

Concept of Queue as an ADT, Representation and Implementation of Linear Queue, Circular Queue, Priority Queue, Double Ended Queue. Applications: Job scheduling, Queue simulation, Categorizing data. Time complexity and Space complexity analysis of Queue operations. Comparison of Linear Data Structures.

Text Books:

References:

Every student should perform 12 to 14 tutorials which will cover topics of all units mentioned in the syllabus of Data Structures and Algorithms I. Tutorial assignments will enhance the understanding of the concepts of problem solving, algorithms and data structures. Students will perform practice exercise on data representation and corresponding implementation of the data structures. Students will get opportunity to develop their logic building abilities.

List of Tutorial Assignments:
Following list of tutorials can be considered as guideline for designing tutorials:
1. Demonstration of C++ program implementation and execution using eclipse tool.
2. Design an algorithm for simple problems like GCD calculation, power calculation etc.
3. Calculate frequency count, time complexity of sample algorithmic constructs.
4. For given algorithms of array operation, write equivalent C++ code.
5. Practice exercise on sorting algorithms for set of predefined inputs.
6. Calculate time complexity of sorting algorithms using concept of frequency count.
7. Practice exercise on searching algorithms for set of predefined inputs.
8. Run through code of searching algorithms.
9. Create a linked list and write algorithms for traversal, delete a node, add node operations on a list.
10. Create a doubly or circular linked list and write algorithms for traversal, delete a node, add a node operations on a list.
11. Solve brain teaser based on recursive code snippets.
12. Demonstration on debugging techniques.
13. Select appropriate data structures and design algorithmic solution to given application.
14. Solve puzzles based on queue data structure.
Course Objectives:
To facilitate the learners
1. To understand Discrete Mathematics concepts and understand its significance in Computer Engineering.
2. To solve problems based on sets, functions and relations.
3. To understand the reasoning and apply it to solve problems.
4. To learn the basic properties of graphs and trees to find solutions of related applications.
5. To learn fundamentals of algebraic systems, permutation and combination.

Course Outcomes:
By taking this course, the learner will be able to
1. Apply contents of Discrete Mathematics in solving problems, formal proofs, reasoning and understand new concepts in Computer Engineering.
2. Solve problems using set, function, relation models and analyse relationship between elements of sets.
3. Understand basic terminologies of graphs and trees and apply it to solve problems on paper.
4. Understand the concepts of groups, rings, permutations and combinations.

Unit 1: Sets and Mathematical Induction (07)

Unit 2: Logic and Propositional Calculus (06)
Introduction to Propositional Logic, Propositions and Compound Propositions, Basic Logic Operations, Propositions and Truth Tables, Tautology and Contradiction, Logical Equivalences, Algebra of Propositions, Conditional and Bi-Conditional Statements, Logical Implications, Predicates and Quantifiers, Nested Quantifiers, First Order Logic, Normal Forms.

Unit 3: Groups, Rings and Permutations and Combinations (08)

Unit 4: Relations and Functions (08)

Unit 5: Graph Theory (07)
Basic Terminology, Multi-Graphs and Weighted Graphs, Sub-Graphs, Isomorphic Graphs, Complete, Regular and Bipartite Graphs, Operations on Graph, Factors of a Graph, Paths and
Circuits, Connectivity, Hamiltonian and Euler Paths and Circuits, Shortest Path in Weighted Graphs (Dijkstra’s Algorithm), Planer Graph and Theorem, Graph Coloring Problem, Travelling Salesman Problem.

Unit 6: Trees


Text Books:

References:

Every student should perform 12-14 tutorials which will cover topics of all units mentioned in the Syllabus of Discrete Mathematics.
Following list of tutorials can be considered as a guideline for designing tutorials in such a way that all topics should be distributed and covered amongst all batches.

List of Tutorial Assignments:
1. Problems on set, multi-set operations and algebra of sets.
2. Problems on Venn diagram.
4. Translating English statement into propositional logic.
5. Translating English statement into predicate logic.
6. Problems on groups.
7. Problems on permutation and combination.
8. Representation of relations and functions, closure of relations and equivalence relation.
13. Solve problems for shortest path in weighted graphs (Dijkstra’s algorithm) : (Paper pencil method)
14. Give paper solution for minimal spanning trees, Kruskal’s and Prim’s algorithms for minimal spanning trees.
CE 2104 DIGITAL SYSTEMS AND COMPUTER ORGANIZATION

Teaching Scheme:
Lectures : 3Hrs/Week
Tutorial : 1Hr/Week

Examination Scheme:
In Semester : 50 Marks
End Semester : 50 Marks
Credits : 4

Prerequisite:
1. Basic Electrical and Electronics Engineering II (ES1201)

Course Objectives:
To facilitate the learners
1. To understand the basic digital circuits and logic design.
2. To apply techniques for designing combinational and sequential circuits.
3. To understand the functional components of a computer and its organization.
4. To understand design issues of instructions and instruction pipelining.
5. To understand and classify memory and input/output organizations.

Course Outcomes:
By taking this course, the learner will be able to
1. Understand and apply the knowledge of basic digital circuits and logic design.
2. Recall and apply the knowledge of combinational and sequential digital circuits.
3. Understand the basic building blocks and their coordination in a computer organization.
4. Recall and classify the instructions and operands for a CPU.
5. Understand and compare memory and input/output organizations.

Unit 1: Combinational Circuits
Minimization of Product of Sum(POS) and Sum of Product(SOP) Functions and Realization Using Logic Gates, Introduction to Numbers and Codes, BCD, Gray, Excess-3 and Their Applications, Code Conversion, Integer and Floating Point Number Representation, Signed and Unsigned Numbers, Arithmetic Operations, Introduction to Basic Arithmetic Logical Unit(ALU) and Floating Point Unit(FPU).

Unit 2: Combinational Logic Design
Realization of Basic Combinational Functions Like Comparison, Decoding, Multiplexing, Demultiplexing, Design of Half Adder and Full Adder, Design of Half Subtractor and Full Subtractor, BCD Adder, Look Ahead and Carry Generator, Introduction to Carry Propagation Adder, Carry Save Adder.

Unit 3: Sequential Circuits Design
Flip Flops (FFs) and Their Excitation Tables, FF Conversions, Shift Registers, Applications of FFs, Asynchronous and Synchronous Counters, Sequence Generators and Detectors Using Moore And Mealy, Introduction to Algorithmic State Machines (ASM) Charts, Notations, Design of A Simple Controller Using ASM.

Unit 4: Introduction to Computer Organization
Introduction to Computer Organization, Function and Structure of A Computer, Functional Components and Their Interconnection, Register Organization, Number and Size of Registers, General Purpose Registers, Design and Organizational Issues of Registers, Control Unit Organization, Hardwired Vs. Microprogrammed Organization.
Unit 5: Characteristics, Functions and Pipelining of Instructions (07)
Instruction Cycle, Type of Instructions, Types of Operands, Instruction Set Design, Machine Instructions Characteristics, Design Issues of Instructions, Instruction Pipelining, Performance and Hazards of Pipelining, RISC, CISC.

Unit 6: Memory and Input/Output Organization (07)

Text Books:

References:

Web References:
1. NPTEL series – nptel.ac.in/courses/117105080/ (Digital System Design by Prof. D. RoyChoudhary, Dept. of Computer Science and Engineering, IIT Kh.)
2. Online Chapters – WilliamStallings.com/COA/COA8e.html

The subject Digital Systems and Computer Organization is a blend of two divergent subjects like Digital Electronics and Computer Organization. During the tutorial sessions, the students are expected to solve numerical problems based on different concepts of digital electronics. The students are expected to design different combinational and sequential circuits. The students will be given demonstration of digital designs implemented on digital boards.

List of Tutorial Assignments:
2. Demonstrate half adder and full adder using digital board.
3. Problem solving for number system and code conversions.
4. Problems on BCD operation to understand integer arithmetic.
5. Demonstration of realization of BCD adder using IC 7483.
6. Problem solving of Boolean expression using multiplexer/demultiplexer.
8. Demonstration of open source software tools used in digital electronics.
9. Design of flip flop conversion such as D to JK, JK to D and JK to T.
10. Design a sequence detector for a given binary sequence using Moore and Mealy methods.
11. Demonstration of components of computer system and their interconnections.
12. Identification of addressing modes of x 86 family for given instructions.
13. Group discussion on applications of RISC and CISC architecture.
CE 2105 PRINCIPLES OF PROGRAMMING LANGUAGES
LABORATORY

Teaching Scheme
Practical : 4 Hrs/Week

Examination Scheme
In Semester : 25 marks
Oral : 25 marks
Credits : 2

Course Objectives:
To facilitate the learners
1) To explore the principles of object oriented programming.
2) To apply object oriented programming concept for developing applications using Java.
3) To apply ArrayList as a Java collection framework for simple application development.
4) To handle built-in and user defined exceptions.
5) To explore functional language programming in python using simple examples.

Course Outcome:
By taking this course, the learner will be able to
1) Design and develop computer programs using the object-oriented concepts.
2) Develop programming application using object oriented programming language Java.
3) Use ArrayList as a Java collection framework.
4) Handle exceptions using inbuilt classes and user defined exceptions.
5) Implement functional programming language concepts in python.

A large part of CE 2106 lab would be in understanding the syntax or semantics of languages which fall under various paradigms like Imperative (C++), Object Oriented (C++, Java), Functional and Scripting (Python). Main focus would be on Java programming whereas C++ and Python assignments are of introductory level as an example of programming paradigm. Assignment statements are in brief. Faculty members are encourage to expand problem statements with variations. Assignments can be framed and expanded in such a way that it explores concepts, language constructs, logic of solution and simple application.

List of Assignments:
Group A: (Mandatory)
1. Develop an object oriented program in C++ to create a student information system.
2. Design a user defined abstract data type ‘Complex’ in Java. Write a program to perform arithmetic operations of two complex numbers.
3. Implement the following concepts by constructing suitable classes in Java-
4. Implement the following concepts by constructing suitable classes in Java –
   a. Abstract classes and abstract methods  b. Interfaces.
5. Create an application for a book shop and maintain the inventory of books that are being sold at the shop.
6. Write a Python program to count the number of articles in a given text.

Group B: (Any three)
1. Create User defined exception to check the specific conditions for recruitment system and throw the exception if the criterion does not met in Java.
2. Create a student result database in Java. Calculate the grades of students. Decide a criteria for best student and short-list students who satisfies the criteria.
3. Find appropriate class hierarchy in banking application and implement it.
4. Find suitable class hierarchy in the human resource department of an organization and implement it.
5. Write a Java program to perform String operations.
6. Write a Java program to create an abstract data types like Stack/Set/Queue/List as an interface and implement its methods.
7. Write a Python program for sorting students marks.

**Group C: (Any one)**
1. Design and develop the Game using Java using Applet (e.g. Tic-Tac-Toe).
2. Write a Python program that prompts a user to enter a list of words and store in another list only those words whose first letter occurs again within the word (e.g. Baboon). The program should display resulting list.
3. Write a program in Python using functional paradigm for generating two sub-lists of even and odd numbers from given list. Perform addition of individual sub-list and display the result.
CE 2106 DATA STRUCTURES AND ALGORITHMS I LABORATORY

Teaching Scheme
Practical : 4 Hrs/Week

Examination Scheme
In Semester : 25 Marks
End Semester : 25 Marks
Credits : 2

Prerequisite:
1. ES 1202 Fundamentals of Programming Language - II
2. ES 1206 Fundamentals of Programming Language Laboratory - II

Course Objectives:
To facilitate the learners:
1. To develop algorithmic foundations to solve problems.
2. To select and use appropriate linear data structure for a given problem statement.
3. To analyze algorithms using time complexity.
4. To implement sorting and searching algorithms.

Course Outcome:
By taking this course, the learner will be able to:
1. Select and apply linear data structures for given problem.
2. Design and implement solution for given problem.
3. Compare alternative solution by analysing algorithms using time complexity.
4. Implement sorting and searching algorithms.

The laboratory assignments are designed in a set of group A, B and C such that students will be able to design and implement solution for a given problem. Group A assignments are designed in such a way that students will choose appropriate data structures to implement solution of a given problem. All the units of the syllabus of Data Structures and Algorithms II are covered in group B assignments. Some assignments of group B are designed to make students able to implement Abstract Data Type of a data structure and use it for a given application. In group C assignments students will design an algorithmic solution for selected problem using concepts covered in the subject Data Structures and Algorithms II.

The laboratory assignments of group A and B are to be submitted by student individually using C++/JAVA object oriented programming language. Group C assignments may be performed in a group of 2 to 4 students from the same batch. For each assignment program code with sample output is to be submitted as a soft copy. Handwritten write up (Title, Objectives, Problem Statement, Algorithms, and Outcomes) of each assignment is to be submitted by students.

List of Assignments
Group A: (Mandatory)
1. Shopkeeper keep a record for different items purchased by customers on a day. Select appropriate data structure and write a program to perform various operations on given information.
2. Design a system to maintain CSI student branch membership information. Choose appropriate data structure.
3. College Library maintains records of books. Write a program to implement sorting, searching operations on it. Use appropriate data structure.
4. Implement Queue as ADT using linked list or array. Use Queue ADT to simulate 'waiting list' operations of railway reservation system.

Group B: (At least six)
1. Implement permutation and combination based on word problem.
2. In a group of M persons, some people can speak English and some people can speak French. Write program to find union, intersection, difference of given sets.
3. Write a program to represent polynomial equation and perform operations to add and evaluate polynomials.
4. Write a program to perform add, multiply, transpose operations on matrices.
5. Write program to perform various operations on strings.
6. A mobile phone list stores name and contact number in ascending order. Write program to search a contact details of specified name.
7. Write a program to store first year CGPA of students. Use various sorting algorithms to sort data.
8. Implement Doubly Linked List as ADT . Use same ADT to simulate Browser URL application.
9. Implement Singly Linked List as ADT. Use same ADT to simulate deck of cards application.
10. Student’s information along with their percentage is stored in linked list for every division. Generate a combine list of students which is sorted in descending order based on their percentage.
11. Implement Stack as ADT using linked list or array. Use same ADT to check given expression is well formed parenthesized.
12. Implement Stack as ADT using linked list or array. Use same ADT to evaluate given postfix expression.
13. Implement Priority Queue as ADT using linked list or array. Use ADT to simulate pizza parlor order management.
14. Operating system stores N jobs and processing time require to complete each job in data structure. Design a program to simulate the job execution sequence.

**Group C:**
Design a game OR Design a small application to manage library data / medical shop data / College admission data / P.M.P.M.L. bus scheduling data etc. using appropriate data structures.