Autonomous Programme Structure (Modified) of Second Year B. Tech. Computer Engineering Academic Year 2019-2020

S. Y. B. Tech. Computer Engineering Semester – I										
Course Code	Course Title	Teaching Scheme Hours / Week			Examination Scheme				Marks	Credit
		Lecture	Tutorial	Practical	In Semester	End Semester	Oral	Practical		
CE 2101	Principles of Programming Languages	3	0	0	50	50	0	0	100	3
CE 2102	Data Structures and Algorithms I	3	1	0	50	50	0	0	100	4
CE 2103	Discrete Mathematics	3	1	0	50	50	0	0	100	4
CE 2104	Digital Systems and Computer Organization	3	1	0	50	50	0	0	100	4
CE 2105	Principles of Programming Languages Laboratory	0	0	4	25	0	25	0	50	2
CE 2106	Data Structures and Algorithms I Laboratory	0	0	4	25	0	0	25	50	2
HS 2101	Principles of Economics and Finance	3	0	0	50	50	0	0	100	3
AC 2101	Self Expression	0	0	2	0	0	0	0	0	No Credit
	Total	**15**	**3**	**10**	**300**	**250**	**25**	**25**	**600**	**22**
	Grand Total	**28**			**600**				**600**	**22**

AC 2101 -- Audit Course : Self Expression

1. Dance
2. Drawing / Painting / Sketching
3. English Communication Skill
4. Film Appreciation
5. Origami
6. Theater

MKSSS's Cummins College of Engineering for Women, Karvenagar, Pune-411052

DEAN ACADEMICS
MKSSS's Cummins College of Engineering for Women Karvenagar, Pune-411052

Principal
MKSSS's Cummins College of Engg. For Women, Karvenagar, Pune-52.

APPROVED BY
Governing Body Members
MKSSS's Cummins College of Engineering for Women Karvenagar, Pune-411052

MKSSS's Cummins College of Engineering for Women, Karvenagar, Pune-411052

DEAN ACADEMICS
MKSSS's Cummins College of Engineering for Women Karvenagar, Pune-411052

Principal
MKSSS's Cummins College of Engg. For Women, Karvenagar, Pune-52.

APPROVED BY
Governing Body Members
MKSSS's Cummins College of Engineering for Women Karvenagar, Pune-411052

CE 2101 PRINCIPLES OF PROGRAMMING LANGUAGES

Teaching Scheme
Lecture : 3 Hrs/week

Examination Scheme
In semester : 50 marks
End semester : 50 marks
Credits: 3

Prerequisite:
ES 1202 Fundamentals of Programming Languages - II

Course Objectives:
To facilitate the learners :
1) To understand and apply object-oriented principles for application development.
2) To develop programming applications using Java.
3) To understand design concepts for programming languages
4) To analyse various programming paradigms.

Course Outcome:
By taking this course, the learner will be able to :
1) Make use of object-oriented principles for effective programming.
2) Construct simple programs using object-oriented programming language Java.
3) Explore languages design concepts for programming languages
4) Classify different programming paradigms for application development.

Unit 1: INTRODUCTION (06)

Role of programming languages, need to study programming languages, characteristics of a good programming languages, introduction to various programming paradigms. Need of object-oriented paradigm, basic concepts of object oriented programming (OOP), benefits of OOP. General characteristics for OOP, concepts - object, classes, messages, methods. Class Identification, object-oriented as abstract data type. Data abstraction, encapsulation, polymorphism, inheritance, dynamic binding, abstract classes, interfaces, generic class, run time type identification

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 (08)

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, Basics of AWT and Swing.

Unit 4: LANGUAGE DESIGN CONCEPTS (06)

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 (06)

Introduction to procedures, parameter passing methods, lifetime of variables, scope rules: static and dynamic scope, nested scope, procedure call and return, recursive sub-program. Referencing

environment, activation records, storage management , desirable and undesirable characteristics of procedural programming

Unit 6:FUNCTIONAL PROGRAMMING **(08)**

Introduction to functional programming, lambda calculus, ambiguity, free and bound identifiers, reductions, typed lambda calculus, application of functional programming. Functional Programming with python, elements of functional Programming, function declaration, expression evaluation, type checking

Text Books:

1. Roosta S., "**Foundations of Programming Languages**", Thomson, Brooke/Cole, (India Edition) (2009)
2. Herbert Schilt, "**JAVA Complete Reference**", Tata McGraw Hill, (9th Edition), (2014)
3. David Mertz, " **Functional Programming in Python** ", O'Reilly , (1st Edition), (2015)
4. Sethi R., "**Programming Languages concepts & constructs**", Pearson Education, (2nd Edition) (2007)

Reference Books:

1. Sebesta R., "**Concepts Of Programming Languages**", Pearson Education, (10th Edition)(2014)
2. Eckel B., "**Thinking in Java**", Pearson Education, (3rd Edition)
3. T. W. Pratt , "**Programming Languages**", Prentice-Hall Of India, (4th Edition),(2009)
4. Summerfield M, " **Programming In Python 3: A Complete Introduction To The Python Language",** Pearson Education. (2nd Edition) (2011).
5. Lutz M, " **Programming Python**", SPD/O'reilly, (4th Edition),(2015).
6. Allen Tucker, Robert Noonan, "**Programming Languages: Principles and Paradigms**", Tata McGraw Hill, (2nd edition),(2007)
7. Carlo Ghezzi, Mehdi Jazayeri, "**Programming Language Concepts**",3rd Ed, Wiley Publication ISBN : 978-81-265-1861-6.

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 learn, apply and analyze various data searching and sorting techniques.
4. To analyze algorithms using time and space complexity.

Course Outcomes:
By taking this course, the learner will be able to:

1. Apply appropriate linear data structure to construct efficient algorithms to approach the

given problem .

2. Apply the concept of Linked list to solve given problem.
3. Distinguish between various linear data structures based on their representations and applications.
4. Solve examples using data searching and sorting techniques.
5. Apply and analyse algorithms using time and space complexity.

Unit 1: Introduction to Algorithm, Data Structures and Analysis of Algorithms (07)
Concept of Problem Solving, Introduction to Algorithms, Characteristics of Algorithms, Pseudo code and Flowchart , Abstract Data Types (ADT), Set as an ADT. Introduction to Data Structures, Classification of Data Structures. Frequency Count, Analyzing Algorithm using Frequency count, Time complexity and Space complexity of an Algorithm, Asymptotic notations, Best, Worst and Average case analysis of an Algorithm.

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. Sparse Matrices

Unit 3: Linked List (08)
Concept of Linked List, Comparison of Sequential and Linked Organizations, Linked List using Dynamic Memory Management, Linked List as an ADT, Introduction to types of Linked List, Linked List operations. Time complexity analysis of Linked List operations. Application: Polynomial as ADT using Linked List.

Unit 4: Stacks (07)

Stack as an ADT, Representation and Implementation of Stack using Sequential and Linked Organization. Applications of Stack- Simulating Recursion using Stack, Arithmetic Expression Conversion and Evaluation, Reversing a String. Time complexity analysis of Stack operations.

Unit 5: Queues **(06)**

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 analysis of Queue operations. Comparison of Linear Data Structures.

Unit 6: Sorting and Searching Techniques **(08)**

Need of Sorting and Searching, Sorting Order and Stability in Sorting. Concept of Internal and External Sorting. Bubble Sort, Insertion Sort, Selection Sort, Quick Sort and Merge Sort, Radix Sort, Shell Sort. Time complexity analysis of Sorting Algorithms. Linear Search, Binary Search, Time complexity analysis of Searching Algorithms.

Text Books:

1. E. Horwitz , S. Sahani, D. Mehta, **"Fundamentals of Data Structures in C++"**, *University Press*, (2nd edition) (2008).
2. R. Gilberg, B. Forouzan, **"Data Structures: A Pseudocode approach with C++"**, *Brooks* (1st Edition) (2001).

References:

1. Yedidyah Langsam, Moshe J Augenstein, Aron M Tenenbaum, **"Data Structures using C and C++"** , *Pearson Education*, (2nd edition) (2009).
2. A. Aho, J. Hopcroft, J. Ulman, **"Data Structures and Algorithms"**, *Pearson Education*, (2nd edition) (2008) .
3. Brassard and Bratley, **"Fundamentals of Algorithmics"**, *Prentice Hall India/Pearson Education*, (2nd edition) (2009).
4. Goodrich, Tamassia, Goldwasser, **"Data Structures and Algorithms in C++"**, *Wiley publication*, (2nd edition) (2011).
5. R. Gillberg, B. Forouzn, **"Data Structures: A Pseudocode approach with C"**, *Cenage Learning*, (2nd edition) (2003).
6. M. Weiss, **"Data Structures and Algorithm Analysis in C++"**, *Pearson Education*, (4th edition) (2002).

List of the Tutorial Assignments:

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.

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 a 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.

CE 2103 DISCRETE MATHEMATICS

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

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

Course Objectives:
To facilitate the learners

1. To understand Discrete Mathematics concepts and its significance in Computer Engineering.
2. To understand set theory, logic and apply reasoning to solve problems.
3. To solve problems based on algebraic systems, functions and relations.
4. To learn the basic properties of graphs, trees and permutation, combination to find solutions of related applications.

Course Outcomes:
By taking this course, the learner will be able to

1. Apply concepts of propositional calculus for solving problems, formal proofs, reasoning and represent problems using concepts of predicate calculus.
2. Solve problems of sets, relation, functions and analyse and justify relationship between elements of sets.
3. Apply the concepts of groups, rings, and analyze properties of groups and rings and solve problems on permutations and combinations.
4. Apply basic terminologies of graphs and trees to solve problems on paper.

Unit 1: Sets and Mathematical Induction **(07)**

Significance of Discrete Mathematics in Computer Engineering, Sets, Subset, Universal Set, Empty Set, Algebra of Sets and Duality, Operations on Sets, Finite and Infinite Sets, Un-Countably Infinite Sets, Multi-Sets, Power Set, Venn Diagram, Principle of Inclusion and Exclusion, Principle of Mathematical Induction.

Unit 2: Logic and Propositional Calculus **(06)**

Propositions, Logical connectives, Conditionals and Bi-Conditionals, Tautology, Contradiction, Contingency, Truth Tables, Logical Equivalences, Algebra of Propositions, Logical Implications, , Conjunctive and disjunctive Normal Forms, Rules of Inference, Predicates and Quantifiers, Nested Quantifiers.

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

Algebraic Systems, Groups, Semi Groups, Monoids, Subgroups, Introduction to Isomorphism, Homomorphism and Automorphism of groups, Cosets and Normal Subgroups, Rings, Integral Domain and Field, Introduction to Permutations and Combinations.

Unit 4: Relations and Functions **(08)**

Introduction to Relations, Product Sets, Pictorial Representation of Relations, Composition of Relations, Closure of Relations, Properties of Binary Relations, Equivalence Relations and Partitions, Partial Ordering Relations, Hasse Diagram, Lattices, Chains and Anti-Chains, Warshall's Algorithm, Functions, Composition of Functions, Invertible Functions, Introduction to Discrete Numeric Functions, Generating Functions and Recurrence Relation.

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 **(06)**

Basic Terminologies in Trees and Properties of Trees, Binary Search Trees, Tree Traversal, Spanning Trees, Fundamental Circuits and Cut Sets, Minimal Spanning Trees, Kruskal's and Prim's Algorithms for Minimal Spanning Trees.

Text Books:

1. Kenneth H. Rosen, **"Discrete Mathematics and its Applications",** 7th Edition, 2012, *Tata McGraw-Hill*, ISBN 978-0-07-338309-5.
2. C. L. Liu and D. P. Mohapatra, **"Elements of Discrete Mathematics"**, 4th Edition, *Tata McGraw-Hill*, 2017, ISBN 978-1-25-900639-5.

References:

1. Norman L. Biggs, **"Discrete Mathematics"**, Second Edition, *Oxford University Press*, 2004, ISBN 0–19 –850717 – 8.
2. J. P.Tremblay and R.Manohar, **"Discrete Mathematical Structures with Applications to Computer Science"**, 1997, *Tata McGraw-Hill*, ISBN 0-07-463113-6.
3. E. Goodaire and M. Parmenter, **"Discrete Mathematics with Graph Theory"**, third edition, *Pearson Education*, 2008, ISBN 81 – 7808 – 827 – 4.
4. B. Kolman, R. Busby and S. Ross, **"Discrete Mathematical Structures"**, 6th Edition, *Pearson Education*, 2009, ISBN 81-7808-556-9.
5. N. Deo, **"Graph Theory with application to Engineering and Computer Science"**, Eastern Economy Edition, *Prentice Hall of India*, 1990, 0 – 87692 – 145 – 4.
6. Seymour Lipsehutz and Marc Lars Lipson **"Discrete Mathematics"**, 3rd Special Indian Edition, ISBN-13: 978-0-07-060174-1.

List of the Tutorial Assignments:

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.

1. Problems on set, multi-set operations,Venn diagram. and algebra of sets.
2. Problems on Principle of Inclusion-Exclusion and Mathematical Induction.
3. Translating English statement into propositional logic and predicate logic.
4. Problems on groups.
5. Problems on permutation and combination.
6. Representation of relations and functions, closure of relations and equivalence relation.
7. Problems on partitions, posets, Hasse diagram and Lattices.
8. Problems on Warshall's Algorithm.
9. Problems on composition of functions, invertible functions, recurrence relation.
10. Problems on multi-graphs and weighted graphs, sub-graphs, isomorphic graphs.
11. Solve problems for shortest path in weighted graphs (Dijkstra's algorithm) : (Paper pencil method)
12. Give paper solution for minimal spanning trees, Kruskal's and Prim's algorithms for minimal spanning trees.

MKSSS's Cummins College of Engineering for Women Karvenagar, Pune-411052

DEAN ACADEMICS
MKSSS's Cummins College of Engineering for Women Karvenagar, Pune-411052

Principal
MKSSS's Cummins College of Engg. For Women, Karvenagar, Pune-52.

APPROVED BY
Governing Body Members
MKSSS's Cummins College of Engineering for Women Karvenagar, Pune-411052

CE 2104 Digital Systems and Computer Organization

Teaching Scheme:
Lectures: 3 Hrs./Week
Tutorial: 1 Hr./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. Make use of the knowledge of basic digital circuit elements and model the logic circuits.
2. Build simple combinational and sequential digital circuits.
3. Justify the use of basic building blocks and their coordination in computer organization.
4. Classify the instructions and operands of x86 CPU.
5. Model and compare memory and I/O organization.

Unit Title	**No. of Lectures**
Unit – I: 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.	**(08)**
Unit – II: 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.	**(06)**
Unit – III: Sequential Circuit's Design Flip flops (FF) and their excitation tables, FF conversions, Shift registers, Applications of FFs, Asynchronous and Synchronous counters, Sequence detectors using Moore and Mealy, Introduction to Algorithmic State Machines (ASM) charts, notations, design of a simple controller using ASM.	**(07)**

Unit – IV: Introduction to Computer Organization **(07)**

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- V: 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 – VI: Memory and Input/output Organization **(07)**

Memory devices and organization, ROM, RAM, EPROM, Flash memory. Cache memory organization, principles, cache design elements, performance characteristics, External memory devices and organization, hard disk, RAID, Introduction to buses, types of buses, bus organization, DMA organization, need, working principle.

Text Books:

1. R. P. Jain, '**Modern Digital Electronics**', *Tata McGraw-Hill*, (3rd Edition), (2003)
2. C. Hamacher, Z. Vranesic and S. Zaky, '**Computer Organization**', *McGrawHill*, (2002)
3. W. Stallings, '**Computer Organization and Architecture - Designing for Performance**', *Prentice Hall of India*,(8th edition), (2002)

Reference Books:

1. Anil Maini, '**Digital Electronics: Principles and Integrated Circuits**', *Wiley India Ltd*, (2008)
2. Malvino, D. Leach, '**Digital Principles and Applications**', *Tata Mc-Graw Hill*, (5th edition)
3. John P Hays, '**Computer Architecture and Organization**', *McGraw-Hill Publication*, (3rd Edition), (2001)
4. Tanenbaum, '**Structured Computer Organization**', *Pearson*, (5th Edition)

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

List of Tutorials

Sr. No	Topic
1	SOP and POS examples
2	Problem solving using K map
3	Problems based on Number system and Code conversions
4	Design of Multiplexer / Demultiplexer
5	Design of Encoder / Decoder
6	Flip Flop conversions
7	Simple problems for sequence detector using Moore, Mealy
8	Design of simple sequential circuits
9	Counter Designing
10	Addressing modes of 8086
11	Cache memory problems.

CE 2106 PRINCIPLES OF PROGRAMMING LANGUAGES LABORATORY

Teaching Scheme
Lecture : 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 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) Develop programming application using object oriented programming language Java
2) Make use of Java collection framework for effective programming.
3) Handle exceptions using inbuilt classes and user defined exceptions
4) 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 Object Oriented (Java), and Functional and Scripting (Python). Main focus would be on Java programming whereas 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. Design a user defined abstract data type 'Complex' in Java. Write a program to perform arithmetic operations of two complex numbers
2. Implement the following concepts by constructing suitable classes in Java - a. Constructors b. Constructor Overloading c. Function Overloading d. Function Overriding e. Inheritance
3. Implement the following concepts by constructing suitable classes in Java - a. Abstract classes and abstract methods b. Interfaces
4. Create an application for a book shop and maintain the inventory of books that are being sold at the shop.
5. Java program to create a simple calculator with basic +, -, /, * using java swing elements.
6. Write a Python program to count the number of articles in a given text.

Group B: (Any Four)
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. 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
2. 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
Practical : 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 linear data structures for given problem.
2. Develop the solution for the given problem using programming language.
3. Analyze solutions using time complexity.
4. Design a small application using linear data structure.

List of Assignments

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.

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. A 'concordance List' is an alphabetical list of words that appear in the book . Implement concordance list using ordered Linked List with insertion function that restrict duplicate value to be inserted in the list.
11. Implement Singly Linked List as ADT. Use it to simulate banking operations.
12. 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.
13. Implement Stack as ADT using linked list or array. Use same ADT to check given expression is well formed parenthesized.
14. Implement Stack as ADT using linked list or array. Use same ADT to evaluate given postfix expression.
15. Implement Priority Queue as ADT using linked list or array. Use ADT to simulate pizza parlor order management.
16. 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.

MKSSS's Cummins College of Engineering for Women Karvenagar, Pune-411052

DEAN ACADEMICS
MKSSS's Cummins College of Engineering for Women Karvenagar, Pune-411052

Principal
MKSSS's Cummins College of Engg. For Women, Karvenagar, Pune-52.

APPROVED BY
Governing Body Members
MKSSS's Cummins College of Engineering for Women Karvenagar, Pune-411052

S. Y. B. Tech. Computer Engineering Semester – II										
Course Code	Course Title	Teaching Scheme Hours / Week			Examination Scheme				Marks	Credit
		Lecture	Tutorial	Practical	In Semester	End Semester	Oral	Practical		
BSCE 2201	Engineering Mathematics III	3	1	0	50	50	0	0	100	4
CE 2201	Data Structures and Algorithms II	3	1	0	50	50	0	0	100	4
CE 2202	Fundamentals of Computer Networks	3	1	0	50	50	0	0	100	4
CE 2203	Operating Systems	3	0	0	50	50	0	0	100	3
CE 2204	Microprocessor Architectures	3	1	0	50	50	0	0	100	4
CE 2205	Data Structures and Algorithms II Laboratory	0	0	4	25	0	0	25	50	2
CE 2206	Microprocessor Architectures Laboratory	0	0	2	25	0	0	0	25	1
CE 2207	Operating Systems Laboratory	0	0	4	25	0	25	0	50	2
	Total	**15**	**4**	**10**	**325**	**250**	**25**	**25**	**625**	**24**
	Grand Total	**29**			**625**				**625**	**24**

MKSSS's Cummins College of Engineering for Women Karvenagar, Pune-411052

DEAN ACADEMICS
MKSSS's Cummins College of Engineering for Women Karvenagar, Pune-411052

Principal
MKSSS's Cummins College of Engg. For Women, Karvenagar, Pune-52.

APPROVED BY
Governing Body Members
MKSSS's Cummins College of Engineering for Women
Karvenagar, Pune-411052

BSCE- 2201 ENGINEERING MATHEMATICS III

Teaching Scheme
Lecture : 3 Hrs./week
Tutorials: 1Hr/week

Examination Scheme
In semester : 50 marks
End semester : 50 marks
Credits : 04

Prerequisite:

1. First order linear ordinary differential equations.
2. Basics of Vector Algebra
3. Integration – basic properties, standard results, Beta & Gamma Functions.
4. Partial Fractions.
5. Permutation & Combination. Basics of probability.

Course Objectives: Mathematics is a necessary path to scientific knowledge which opens new perspective of mental activity. Our aim is to provide sound knowledge of engineering mathematics to make the students think mathematically and strengthen their thinking power to analyze and solve engineering problems in their respective areas.

Course Outcome: Students will be able to

CO1. Solve Higher order Linear differential Equations , Simultaneous Differential Equations.
CO2. Calculate Divergence, Curl, Directional derivative, Solenoidal, Irrotational, Scalar potential, vector identities, Line integral.
CO3. Find Fourier Transform, Inverse Fourier Transform.
CO4. Find Z-transforms, Inverse Z – Transform, difference equation.
CO5. Calculate Moments, Mean, Variance, Covariance ,Correlation, Probability Distributions, Compute Skewness, Kurtosis, Linear Regression.

Unit 1: Higher Order Linear Differential equation and application **(07)**

Higher order Linear differential Equation with constant coefficients, Cauchy's and Legendre's Differential Equations, Simultaneous Differential Equations, Modelling of electrical circuits.

Unit 2: Vector Calculus **(05)**

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

Unit 3: Fourier Transform **(05)**

Complex exponential form of Fourier series, Fourier integral theorem, sine and cosine integrals, Fourier transform, Fourier Sine and Cosine transform , Inverse Fourier Transform, Discrete Fourier Transform.

Unit 4: Z – Transform **(06)**

Definition, standard properties, Z- Transform of standard sequences, Inverse Z – Transform using standard results, Inversion integral method, solution of difference equation to solve Computer Engineering Problems.

Unit 5: Probability **(07)**

Theorem of total probability, Theorem of Compound Probability, Baye's theorem, Moments, Mean, dispersion,Variance, Covariance ,Correlation, Random variables, Distributions – Binomial , Poisson, Normal .

Unit 6: Data Analytic **(05)**

Types of data: Concepts of population and sample, quantitative & qualitative data, cross-sectional and time-series data, discrete and continuous data, Skewness, Kurtosis, Linear Regression.

Text Books:

1. B. S. Grewal, **'Higher Engineering Mathematics '**, *Khanna Publications.*
2. B. V. Ramana, **'Higher Engineering Mathematics '**, *Tata McGraw Hill Publications* (2007)
3. C.R.Wylie, L.C. Barrette, '**Advanced Engineering Mathematics**', *McGraw Hill Publications, New Delhi.(*6th edition)(2003)

References:

1. Peter V. O'neil,'**Advanced Engineering Mathematics**' ,*Thomson Brooks / Cole, Singapore (*5th edition) (2007).
2. Erwin Kreyszig ,'**Advanced Engineering Mathematics**' *Wiley Eastern Ltd.* (8th Student Edition), (2004).
3. S.P.Gupta ' **Statistical Methods**', S.Chand & sons

List of the Tutorials:

1. LDE, Rules to find C.F. , General method to find P.I, Shortcut Rules to find P.I.
2. Short-cut Rules to find P.I, Method of Variation of parameters.
3. Cauchy's and Legendre's DE, Simultaneous DE, Modelling of electrical circuits.
4. Vector differentiation, Curl, Divergence.
5. Vector identities, Vector Integration.
6. Fourier integral theorem, Fourier transform, Fourier Sine and Cosine transform.
7. Inverse Fourier Transform, Discrete Fourier Transform.
8. Z Transforms - standard properties. Z- Transform of standard sequences, Z- Inverse Transform of standard sequences
9. Z- Inverse Transform (By various methods), Solution of difference equation by Z – Transform .
10. Theorem of total probability, Theorem of Compound Probability, Baye's theorem, Moments, Mean, dispersion,Variance, Covariance.
11. Distributions – Binomial , Poisson, Normal .
12. Skewness, Kurtosis, Linear Regression.

CE 2201 DATA STRUCTURES AND ALGORITHMS II

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

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

Prerequisite:
CE 2102 - Data Structures and Algorithms I

Course Objectives:
To facilitate the learners:

1. To learn and understand representation, implementation and applications of trees, search trees, graphs, multiway trees data structures.
2. To choose and apply data structures for developing solutions for solving problems in various domains.
3. To analyze algorithms using time complexity analysis.
4. To understand and apply the concepts of hashing and file handling.

Course Outcomes:
By taking this course, the learner will be able to

1. Apply appropriate non linear data structure to construct efficient algorithms to approach the problems.
2. Distinguish between various non linear data structure based on their representations and applications.
3. Apply the concept of Hashing techniques for solving a problem.
4. Make use of File handeling and Java collection Frameworks for solving a problem.

Unit 1: Trees **(08)**

Introduction to Non Linear Data Structure, Binary Trees,Types of Binary Trees,Properties of Binary Trees, Binary Tree as Abstract Data Type, Representation using Sequential and Linked Organization, Binary Tree creation , Recursive and Non Recursive Tree Traversals, Threaded Binary Tree and operations, Traversals of Inorder Threaded Binary Tree, Applications of Binary Trees.

Unit 2: Search trees **(08)**

Representation of Symbol Tables- Static Tree Table and Dynamic Tree Table, Binary Search Tree and its operations, Binary Search Trees as Abstract Data Type, Height Balanced Tree : AVL Tree and operations ,Red Black Tree.

Unit 3: Graphs **(07)**

Basic Terminologies, Storage Representation, Graph Traversals, Graph as Abstract Data Type, Spanning Trees, Minimum Spanning Trees, Kruskal's Algorithm, Prim's Algorithm, Dijkstra's Single Source Shortest Path Algorithm,Topological Sorting, Case Study : Data structures used in Google map.

Unit 4: Multiway trees and Heap **(06)**

Multiway search tree , B Tree and operations, B+ Tree,Applications of Btrees, Heap basic concepts, Realization of Heap, Heap as an Abstract Data Type, Heap implementation, Heap Sort, Heap as a Priority Queue.

Unit 5: Hashing **(07)**

General idea of Hashing, Hash Table, Hash function, Rehashing, Issues in Hashing, Collision Resolution Strategies: Linear Probing, Quadratic Probing, Double Hashing, Open addressing and Chaining.

Unit 6: File Organization and Java Collection Framework **(06)**

File Organization, Sequential File, Direct Access File and its Primitive operations, Java Collection Framework : Arraylist , Treeset, Hashset and Hashmap Class.

Text Books:

1. Sartaj Sahani, **"Data Structures, Algorithms and Applications in JAVA",** *Universities Press* (2nd edition), (2007).
2. Robert Lafore , **"Data Structures Algorithms in JAVA"**, *Techmedia*,(1st edition), (2006).
3. Ivor Horton, **"Beginning Java"**, *Wiley India Edition*,(Java 7 edition),(2012).
4. E. Horowitz, S. Sahni, D. Mehta, **"Fundamentals of Data Structures in C++"**, *Galgotia Publications* ,(2nd edition), (2008).

References:

1. Sartaj Sahani, **"Data Structures, Algorithms and Applications in C++",** *Universities Press* (2nd edition), (2007).
2. R. Gillberg, B. Forouzn, **"Data Structures: A Pseudo code approach with C++",** *Cenage Learning* (2nd edition) (2007).
3. Y. Langsam, M. Augenstein and A. Tenenbaum, **"Data structures using C and C++",** *Prentice Hall of India* (2nd edition), (2005) .
4. M. Weiss, **"Data Structures and Algorithm Analysis in C++",** *Pearson Education* (3rd edition), (2009).
5. A. Aho, J. Hopcroft, J. Ullman, **"Data Structures and Algorithms"**, *Pearson Education* (3rd Impression), (2008).

List of the Tutorial Assignments:

Every student should perform 12 to 14 tutorials which will cover topics of all units mentioned in the syllabus of Data Structures and Algorithms II. Students will perform practice exercise on data representation and corresponding implementation. Tutorial assignments will help students to enhance their ability of problem solving using appropriate data structures.

Following list of tutorials can be considered as guideline for designing tutorials:

1. Practice exercise on creating a binary tree and perform recursive and non recursive traversals of binary tree on given data.
2. Create a binary search tree for the given data and perform its inorder, preorder, postorder traversals.
3. Practice exercise on searching and deleting data values from given binary search tree. Analyze the time complexity of used algorithm.
4. Create a binary search tree for the given data and perform its inorder, preorder, postorder traversals.
5. Practice exercise on different rotations of AVL tree.
6. Construct AVL tree for the given numeric data elements. Perform the appropriate rotations whenever needed.
7. Simulate flight path data using graph data structure to find minimum cost path.
8. Practice assignment on converting a binary tree to threaded binary tree and its traversals.
9. Design a heap data structure for student data and find out minimum/maximum marks obtained in particular subject.
10. Use sequential file to maintain employee information. Write algorithm to add, delete and search employee information from the file.
11. Design a solution for company survey about its products in an area. Choose the appropriate algorithm to complete the survey within short time period and cover all

houses under that area. Give justification for your answer and also analyze your algorithm for time complexity.

12. Given the input data and hash function , show the result using following hashing methods

    a. Linear Probing b. Quadratic Probing c. Double hashing h2(x)=7-(x Mod 7)

13. Use different hashing functions to hash given values.
14. Construct a Btree of order 3 by inserting numbers of given data.

CE2202 FUNDAMENTALS OF COMPUTER NETWORKS

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

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

Course Objectives:
To facilitate the learners

1. To learn and understand fundamental concept of networking.
2. To learn different methods for framing, flow control, error control.
3. To understand OSI model & TCP/IP protocol stack.
4. To learn various functions of physical & data link layer.

Course Outcomes:
By taking this course, the learner will be able to

1. **Build** the knowledge of fundamental concepts of networking to recognize various network standers and protocols.
2. **Build** the knowledge of design requirements of layered network architecture.
3. **Analyze** different error and flow control strategies.
4. **Experiment with** different line coding techniques, modulation techniques and switching techniques to build design requirements of physical layer.

Unit 1: Introduction to Computer Networks **(08)**

Concept of Data in Networking-Representation, Transmission, Data Flow, types of Connection- Point to Point ,Point to Multi Point , Network Standards, type of Networks-LAN,WAN,MAN, Ad-hoc Network, Networking Topologies: Bus, Mesh, Star, Ring and Hierarchical , The Internet-dial up,DSL service, Internet Standards, Internet administration.

Unit 2: Network Models **(06)**

Principles of protocol layering,The TCP/IP Protocol suite:Layers, description of each layer , encapsulation and decapsulation, addressing,multiplexing and demultiplexing, OSI Model, OSI verses TCP/IP suite.

Unit 3: Physical Layer **(08)**

Digital signals,Digital to digital conversion:line Coding techniques(unipolar,polar and bipolar), analog to digital conversion:(PCM, DM),Transmission modes:parallel,serial, introduction to Multiplexing and types: FDM, TDM,Transmission Media- Guided(Twisted pair cable, coaxial cable, Fiber Optic),Unguided media:propagation methods, types of waves (radio waves,microwaves and infrared waves) .Introduction to Switching-Circuit Switching, Packet Switching, Message Switching.

Unit 4: Logical Link Control Sublayer (LLC) **(08)**

Design issues, services, functions, Framing, Error Control and Flow Control, Error Control-Parity Bits, Hamming Code & CRC, Flow Control-Unrestricted Simplex Protocol, Stop and Wait, Sliding Window Protocol.

Unit 5: Medium Access Control Sublayer (MAC) **(08)**

Channel Allocation-Static and Dynamic, Multiple Access Protocols: CSMA, , IEEE

802.Standards and Frame Formats, CSMA/CD, CSMA/CA.

Unit 6: Connecting Devices and Virtual LAN **(04)**

Various Network Devices NIC, Switches, Hub, Routers, Repeaters, Bridge and Access Point. Virtual LANS : membership, configuration, communication between switches, advantages.

Text Books:

1. Fourauzan B., "Data Communications and Networking", 5th Edition, Tata McGraw- Hill, Publications, 2006.
2. William Stallings "Data and computer communication", Pearson, 8th Edition, ISBN: 0-13-243310-9

References:

1. Kurose, Ross "Computer Networking a Top Down Approach Featuring the Internet",6th edition (March 5, 2012), Pearson , ISBN-10: 0132856204.
2. Andrew S. Tenenbaum, "Computer Networks", 5th Edition, PHI, ISBN 81-23-2175-8.

MKSSS's Cummins College of Engineering for Women Karvenagar, Pune-411052

DEAN ACADEMICS
MKSSS's Cummins College of Engineering for Women Karvenagar, Pune-411052

Principal
MKSSS's Cummins College of Engg. For Women, Karvenagar, Pune-52.

APPROVED BY
Governing Body Members
MKSSS's Cummins College of Engineering for Women Karvenagar, Pune-411052

List of the Tutorial Assignments:

Every Student should perform 12-14 tutorials which will cover topics of all units mentioned in the syllabus of Fundamentals of Computer Network.

Tutorial assignments will help students learn and explore the subject in greater detail. Students will be able to recall and practically apply the concepts learnt. Students will emulate algorithms to get insight of the strategies used for flow control.

1. Basic concepts of Computer Networking.
2. Execute and understand basic Networking Commands.
3. Study and discuss various Network components, devices and Structured Cabling components.
4. Problems on Line Coding techniques- POLAR (RZ, NRZ)
5. Problems on Line Coding techniques- Polar Biphase: Manchester and Differential Manchester encoding
6. Problem solving on Error Control coding through Hamming code technique.
7. Problem solving based on CRC technique.
8. Problem solving on basic Flow Control strategies: Sliding Window protocol (Go Back N).
9. Problem solving on basic Flow Control strategies: Sliding Window protocol (Selective repeat).
10. Create Peer to Peer network and LAN network to share files within the created network.

CE 2203 OPERATING SYSTEMS

Teaching Scheme
Lectures : 3 Hrs/week

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

Prerequisites:

1. Fundamentals of Programming Languages – II (ES 1202)
2. Digital Systems and Computer Organization (CE 2104)

Course Objectives:
To facilitate the learner -

1. To understand basic concepts of Operating Systems.
2. To understand process life-cycle and scheduling algorithms.
3. To analyze memory management strategies.
4. To understand File System concepts.
5. To learn operating system for managing resources such as I/O, CPU, memory etc.
6. To understand Inter-process Communication and deadlock concepts.

Course Outcomes:
By taking this course, the learner will be able -

1. To Build the basic knowledge of operating system.
2. To Apply the process concepts and compare the CPU scheduling algorithms.
3. To Analyze memory management strategies.
4. To Apply the file attributes and different access modes on various types of file.
5. To Make use of the knowledge of storage devices for disk management.
6. To Examine the concepts of Inter-process Communication.

Unit 1: Introduction to Operating Systems **(06)**

Introduction to Operating System (OS), Evolution of OS, Functions of OS, Types of OS, OS Concepts, Process, Files, Shell and its types, Kernel and its types, System Calls, Virtual Machine, Case Study of UNIX Operating System.

Unit 2: Process and CPU Scheduling **(08)**

Process Concept, Operations On Processes, Creation, Termination, States, Transition and Context Switching, Scheduling Criteria, Scheduling Algorithm, First-Come First-Serve (FCFS), Shortest Job First (SJF), Round-Robin (RR), Introduction to Threads and Benefits, Case Study of Unix Process Management.

Unit 3: Memory Management **(08)**

Contiguous and Non-Contiguous Memory, Swapping, Paging, Segmentation, Virtual Memory, Demand Paging, Page Replacement Algorithms- First-In First-Out (FIFO), Least Recently Used (LRU), Optimal, Allocation of Frames and Trashing.

Unit 4: Introduction to the File System **(06)**

File Concepts, File Attributes, File Operations, File Types, File Sharing, File Structure, Mounting and Un-Mounting, Directory Overview, Types of Directories, Types of Users, Access Modes, Free space management, Case Study of UNIX File Structure.

Unit 5: I/O Management and Disk Scheduling **(06)**

I/O Devices, Organization of I/O Functions, Operating System Design Issues Related to I/O, I/O Buffering, Disk Scheduling - First Come-First Serve (FCFS), SCAN, Circular SCAN (C- SCAN), Shortest Seek Time First (SSTF).

Unit 6: Inter-Process Communication (IPC) **(08)**

Critical Section Problem, Hardware Support for Mutual Exclusion, Semaphores, Classical Problems of Synchronization, Monitors, Deadlocks, Methods of Handling Deadlock, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection and Recovery from Deadlock.

Text Books:

1. William Stallings, "Operating System-Internals and Design Principles ", Prentice Hall India,(5/e) ISBN: 81-297-0 1 094-3.
2. Silberschatz, Galvin, Gagnes, "Operating System Concepts", John Wiley & Sons, (6/e), ISBN: 9971-51-388-9.
3. Maurice J. Bach, "The Design of the Unix Operating System", Pearson Education, ISBN: 81-7758-770-6.

References:

1.Evi Nemeth, Garth Snyder, Tren Hein, Ben Whaley, "Unix and Linux System Adminstration Handbook", (4/e), ISBN: 978-81-317-6177-9. (2011).
2. Milan Milenkovic, "Operating Systems", TMH,(2/e), ISBN: 0-07-044700-4.
3. Andrew S. Tanenbaum, "Modern Operating Systems", Prentice Hall India, (2/e), ISBN: 81-203-2063-8.

CE 2204 MICROPROCESSOR ARCHITECTURE

Teaching Scheme:
Lectures: 3 hrs./Week
Tutorial: 1 hr./Week

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

Prerequisite:

1. Digital Systems and Computer Organization (CE 2104)

Course Objectives:

To facilitate the learners

1. To understand basic architecture of 8086 microprocessor.
2. To understand and analyze the basic interfacing techniques.
3. To understand pipelined and superscalar architecture of Pentium.
4. To understand, apply and analyze x86 microprocessor instructions to the assembly language programming.

Course Outcomes:

By taking this course, the learner will be able to

1. Recall and apply the knowledge of 8086 microprocessor and apply the programming concept for x86 assembly level language.
2. Understand and analyze the different components and peripherals associated with microprocessor architectures.
3. Recall and understand the advanced features of Pentium microprocessor.
4. Understand advanced microprocessor architectures and their design issues.

Unit – 1: BASIC MICROPROCESSOR ARCHITECTURE **(06)**

8086 Architecture, Pin diagram of 8086, Programmers' model of 8086, pin Diagram, Addressing Modes, Instruction Set, Memory architecture of 8086, Segmentation, even and odd memory banks, address mapping.

Unit – 2: BASIC INTERFACING TECHNIQUES **(06)**

Block diagram, control words, operating modes, programs of Parallel peripheral interface with 8255 (Programmable Peripheral Interface), Block diagram, control words, operating modes, programs of Serial peripheral interface with 8251 (USART), Block diagram, control words, operating modes, programs of Timing and control signals handling using 8253 (Programmable Interval Timer).

Unit – 3: SUPERSCALAR ARCHITECTURE IN PENTIUM MICROPROCESSOR **(06)**

Pentium Architecture, Pipeline stages, Superscalar pipeline issues, Instruction paring rules, Branch prediction, Memory organization with Instruction and Data caches Pentium programmers' model, Register set, Addressing modes and instructions (other than 8086).

Unit – 4: PROTECTED MODE ARCHITECTURE IN PENTIUM MICROPROCESSOR **(06)**

Real Mode vs. Protected mode, Memory management with segmentation and paging Protection mechanism in segmentation and paging, Virtual 8086 Mode (support registers, descriptors, privilege-level, protection, exclusive instructions, inter-privilege level, transfer control, Paging-support

registers, Descriptor, linear to physical address translation, TLB, page level protection).

Unit – 5: MULTITASKING, INTERRUPTS, EXCEPTION AND INPUT/OUTPUT (06)
Multitasking, support registers, Descriptors, Task switching, Nested task, I/O handling in Pentium, I/O instructions, I/O Permission bit map, Interrupts and Exceptions structure in real, protected and virtual modes.

Unit–6: INTRODUCTION TO ADVANCED MICROPROCESSOR ARCHITECTURES (06)
Introduction to multicore architectures i3/i5/i7, Design Issues, Cache coherency Advanced Processor Architectures for Mobile Application, Embedded Application and Enterprise Application.

Text Books:

1. 8086 and peripherals – Intel Manual
2. Pentium Architecture – Intel Manual
3. Douglas Hall, **'Microprocessors & Interfacing'**, *McGraw Hill*, (Revised 2nd Edition), (2006)
4. James Antonakos, **'The Pentium Microprocessor'**, *Pearson Education,* (2nd Edition), (2004)

Reference Books:

1. Sivarama P. Dandamudi, **'Introduction to Assembly Language Programming For Pentium and RISC Processors'**, *Springer*,(2nd Edition), (2004)
2. Peter Abel, **'Assembly language programming'**, *Pearson Education*,(5th Edition),(2002)
3. John Uffenbeck, **'The 8086/88 Family: Design, Programming & Interfacing'**, *PHI*, (2nd Edition), (2002)
4. A. Ray, K.Bhurchandi, **'Advanced Microprocessors and peripherals: Architecture, Programming & Interfacing'**, *Tata McGraw Hill*, (2nd Edition), (2004)
5. Liu, Gibson, **'Microcomputer Systems:The8086/88Family'**,*PHI*,(2nd Edition),(2005)
6. Kip Irvine, **'Assembly language for IBM PC'**, *PHI*, (2nd Edition),(1993)

Web References:

1. NPTEL series – nptel.ac.in/courses/Webcourse-contents/IIT-KANPUR/
2. service.scs.carleton.ca/sivarama/org_book/org_book_web/slides/chap_1_versions/ch7_1.pdf

List of Tutorial Assignments:

The subject Microprocessor Architectures introduces the processor evolution from basic to advance. The tutorial is designed to develop assembly language programming ability of an individual student.

1. & 2. Write small code snippets using arithmetic, logical and conditional jump instructions.
3. Learning how to use DOS/LINUX system calls for program I/O.
4. Numerical examples solving of logical to physical address translation for x86.
5. Write small codes using string instructions.
6. Design of Delays through programming in assembly language.
7. Evaluate the output of small ALP's.
8. Draw memory maps and evaluate the changes after a particular instruction.
9. Handling of ASCII x86 instructions.
10. Handling of BCD x86 instructions.
11. Open book test for addressing modes of Pentium.
12. Write small program using floating point instructions.
13. &14. Group activities related to white papers/presentations/ videos/tools on recent technical processors architecture advancement.

CE 2205 DATA STRUCTURE AND ALGORITHMS- II LABORATORY

Teaching Scheme
Practical : 4 Hrs/Week

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

Prerequisite:
CE2107 - Data Structures and Algorithm Laboratory I

Course Objectives:
To facilitate the learners

1. To choose and apply appropriate Data Structures for a given problem statement.
2. To design algorithmic solution for a given problem.
3. To analyze and compare algorithms.
4. To implement non linear data structures using Object Oriented Programming.

Course Outcome:
By taking this course, the learner will be able to

1. Develop a solution of the given problem using tree data structure.
2. Develop a solution of the given problem using graph data structure.
3. Apply hashing techniques to solve a given problem.
4. Make use of sequential file handling operations.
5. Design small application using non linear data structures.

List of Assignments

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. 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, Outcomes) of each assignment is to be submitted by students.

Group A (Mandatory)

1. Create a Dictionary that stores keywords and its meanings, using appropriate data structure. Implement its operations such as add, delete, display, search and update its values.
2. Create a reasonably balanced tree to maintain names and telephone numbers of all the customers of a shopkeeper and perform operations on it. Test your program for at least 10 names.
3. A news paper delivery boy every day drops news paper in a society having many lanes and houses. Design a program to provide different paths that he could follow. Solve the problem by suggesting appropriate data structures. Design necessary class.
4. Write a program to create telephone book database of N clients. Make use of a hash table implementation to quickly look up client's telephone number.

Group B (At-least Six)

1. Create a binary tree and perform inorder ,preorder and postorder traversals.
2. Implement Binary Search Tree as Abstract Data Type and perform operations on it.
3. Write a program to create a binary tree if inorder and preorder or inorder and postorder any two traversals are given.
4. Create inorder threaded binary tree and perform its traversals.
5. Company wants to lease phone lines to connect its offices of different cities, with each other. Company charges different amounts of money to connect different pairs of offices. Solve the problem by suggesting appropriate data structures to connect all offices of a company with a minimum cost.
6. Write a modular program to implement primitive operations on Min/Max Heap.
7. Write a program to implement Symbol Table as an ADT.
8. Use sequential file to maintain student information. Write algorithm to add, delete and search student information from the file.
9. Implement hash table ADT and handle the collision using linear probing and chaining (with or without replacement). Perform operations on it.

Group C

Create a small application using appropriate data structures to process stock data / organization's data / college data.

CE 2207 MICROPROCESSOR ARCHITECTURES LABORATORY

Teaching Scheme:
Lectures: 2 hrs./Week

Examination Scheme:
In Semester – 25 marks
Credit(s): 1

Prerequisite:

1. Digital Systems and Computer Organization (CE 2104)

Course Objectives:

To facilitate the learners

1. To understand and apply x86 instructions to write assembly language program.
2. To learn, apply and analyze microprocessor and peripherals interfacing techniques.
3. To learn and use the interfacing of assembly language and higher-level language.
4. To able to solve moderately complex problems using modular assembly language programming.
5. To understand and use privileged instructions.

Course Outcomes:

By taking this course, the learner will be able to

1. Recall, classify and apply x86 instructions to write assembly language program.
2. Build a small system using microprocessor interfacing techniques.
3. Solve a given problem using inline coding.
4. Apply the modular programming using assembly level language.

The Microprocessor Architectures laboratory assignments are designed using assembly language programming as well as hardware interfacing techniques. The laboratory work also covers the assembly language interface with higher level language like 'C'. The students are introduced to advanced protected mode instructions.

Group A Assignments (Perform all assignments)

1. Write ALP to perform basic arithmetic operations and check the output in debugger.
2. Write ALP to accept a string and display it on the screen.
3. Write ALP to accept a signed number and check if it is positive or negative. Display appropriate message.
4. Write 8086 ALP to interface DAC and generate following waveforms on oscilloscope
   (i) Square wave – Variable Duty Cycle and frequency.
   (ii) Ramp wave – Variable direction
   (iii) Trapezoidal wave
   (iv) Stair case wave
   (v) Temple wave
   (vi) Sine wave – using look up table
5. Write 8086 ALP to program 8251 for serial communication between two 8251s.
6. Write 8086 ALP to program 8253 to observe outputs of different modes using counter display.
7. Write ALP using STRING instructions to accept a string from user and perform following operations
   (a) Convert a string to uppercase / lowercase

(b) Toggle the case of the string
(c) Concatenation of another string
(d) Find if it is palindrome
(e) Find a substring
(For this assignment make a group of 4 students, each one performing each task and then combine all functions to apply modular programming.)

Group B Assignments (Perform any two)

1. Write ALP to perform following using command line arguments to simulate TYPE or COPY command.
2. Write ALP to find the largest number from an array using PUBLIC/GLOBAL and EXTERN.
3. Write a C/ inline program for PC to PC communication.
4. Write ALP for Mouse interface.
5. Write inline code to perform file operations.
6. Write ALP for floating point operations.

Group C Assignments (Perform any one)

1. Write ALP for to read GDTR/LDTR and IDTR and display the table content pointed by GDTR and IDTR.
2. Write ALP to implement multitasking using Pentium programming.

Text Books:

1. 8086 and peripherals – Intel Manual
2. Pentium Architecture – Intel Manual
3. Douglas Hall, **'Microprocessors & Interfacing'**, *McGraw Hill*, (Revised 2nd Edition), (2006)
4. James Antonakos, **'The Pentium Microprocessor'**, *Pearson Education*, (2nd Edition), (2004)

Reference Books:

1. Sivarama P. Dandamudi, **'Introduction to Assembly Language Programming For Pentium and RISC Processors'**, *Springer*,(2nd Edition), (2004)
2. Peter Abel, **'Assembly language programming'**, *Pearson Education*,(5th Edition),(2002)
3. John Uffenbeck, **'The 8086/88 Family: Design, Programming & Interfacing'**, *PHI*, (2nd Edition), (2002)
4. A.Ray, K.Bhurchandi, **'Advanced Microprocessors and peripherals: Architecture, Programming & Interfacing'**, *Tata McGraw Hill*, (2nd Edition), (2004)
5. Liu, Gibson, **'Microcomputer Systems:The8086/88Family'**,*PHI*,(2nd Edition),(2005)
6. Kip Irvine, **'Assembly language for IBM PC'**, *PHI*, (2nd Edition),(1993)

Web References:

1. NPTEL series – nptel.ac.in/courses/Webcourse-contents/IIT-KANPUR/
2. service.scs.carleton.ca/sivarama/org_book/org_book_web/slides/chap_1_versions/ch7_1.pdf

CE 2207 OPERATING SYSTEMS LABORATORY

Teaching Scheme
Practical : 4 Hrs/week

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

Prerequisites:

1. Data Structures and Algorithms-I (CE2102)
2. Fundamentals of Programming Language Lab-II (ES1206)
3. Digital Systems and Computer Organization(CE2104)

Laboratory Objectives:
To facilitate the learners -

1. To learn and understand the fundamentals of Operating Systems.
2. To learn shell scripting to automate operating system operations.
3. To understand the operations performed by Operating System as a resource manager.
4. To apply the concepts of Operating System for Process and Memory management.
5. To implement and analyze various scheduling algorithms.
6. To explore the communication among the processes.

Laboratory Outcomes:
By taking this course, the learner will be able -

1. To choose and construct Unix/Linux Commands and Shell Programming.
2. To analyze different CPU scheduling algorithms.
3. To apply and analyze Memory Management algorithms.
4. To apply and analyze various disk scheduling algorithms.
5. To examine the Inter-Process Communication concepts.

Every student should perform 9-10 assignments in this laboratory which will cover topics of all units mentioned in the syllabus of Operating Systems. Following is the list of assignments that can be considered as guideline for designing assignments and give basic knowledge of operating systems and its services. The choice of the assignments for each student is given in such a way that all topics should be distributed and covered amongst all batches.

List of Assignments:
Group A: (Mandatory)

1. Write a shell script to implement mount and un-mount commands to mount device and un-mount it.
2. Exploration of Unix/Linux Commands (File, Directory and Process commands).
3. Write a program to implement Banker's Algorithm for deadlock handling.
4. To implement Reader-Writer problem using semaphores.

Group B: (Any four)

5. Write a program to implement following Non- Pre-emptive scheduling algorithms : First Come First Serve (FCFS), Shortest Job First (SJF).
6 .Write a program to implement following Pre-emptive scheduling algorithms: Round-Robin (RR), Shortest Remaining Time First (SRTF)
7. Write a program to implement following memory allocation strategies: First Fit, Best Fit and Worst Fit.
8. Write a program to implement following Page replacement algorithms: a) First-In-First-Out (FIFO). b) Least Recently Used (LRU) c) Optimal page replacement.
9. Write a shell script for adding users / groups and modifying permissions of file / directory accordingly.
10.Write a program to implement following disk scheduling algorithms: First Come First Serve (FCFS), SCAN, Circular - SCAN(C-SCAN), Shortest Seek Time First (SSTF).

Group C: (Any one)

1. Installation of Linux Operating System.
2. Implement producer-consumer algorithm using multi-threading concept.