SCHEME OF INSTRUCTION & EXAMINATION
B.E (Computer Science and Engineering)
SEMESTER-III

S. No.	Course Code	Course Title	Scheme of Instruction				Scheme of Examination			Credits
			L	T	D/P	Contact Hrs/Wk	CIE	SEE	Duration in Hrs	
Theory Courses										
1	ES 301 ME	Applied Operations Research	3	-	-	3	30	70	3	3
2	ES 301 EC	Basic Electronics	3	-	-	3	30	70	3	3
3	PC 301 CS	Data Structures	3	-	-	3	30	70	3	3
4	PC 302 CS	Discrete Mathematics	3	-	-	3	30	70	3	3
5	PC 303 CS	OOP using JAVA	3	-	-	3	30	70	3	3
6	PC 304 CS	Logic and Switching Theory	3	-	-	3	30	70	3	3
7	HS 201 EG	Effective Technical Communication in English	3	-	-	3	30	70	3	3
Practical / Laboratory Courses										
8	PC 351 CS	Data Structures Lab	-	-	2	2	25	50	3	1
9	ES 351 EC	Basic Electronics Lab	-	-	2	2	25	50	3	1
10	PC 352 CS	OOP using JAVA Lab			2	2	25	50	3	1
Total			**21**	**--**	**6**	**27**	**285**	**640**	**30**	**24**

SEMESTER-IV

S. No.	Course Code	Course Title	Scheme of Instruction				Scheme of Examination			Credits
			L	T	D/P	Contact Hrs/Wk	CIE	SEE	Duration in Hrs	
Theory Courses										
1	PC 401 CS	Automata Languages & Computation	3	-	-	3	30	70	3	3
2	HS 406 CM	Managerial Economics and Accountancy	3	-	-	3	30	70	3	3
3	BS 207 MT	Probability & Statistics	3	1	-	4	30	70	3	4
4	PC 402 CS	Operating Systems	3	-	-	3	30	70	3	3
5	PC 403 CS	Computer Organization	3	-	-	3	30	70	3	3
6	ES 401 EC	Signals and Systems	3	-	-	3	30	70	3	3
7	PC 404 CS	Database Management Systems	3	-	-	3	30	70	3	3
Practical / Laboratory Courses										
8	PC 451 CS	Operating Systems Lab	-	-	2	2	25	50	3	1
9	PC 452 CS	Computer Organization Lab	-	-	2	2	25	50	3	1
10	PC 453 CS	Database Management Systems Lab	-	-	2	2	25	50	3	1
Total			**21**	**1**	**6**	**28**	**285**	**640**	**30**	**25**

APPLIED OPERATIONS RESEARCH

ES301ME	**Instruction: 3 periods per week**	**Duration of SEE: 3 hours**
CIE: 30 marks	**SEE: 70 marks**	**Credits: 3**

Objectives:

Use variables for formulating complex mathematical models in management science, industrial engineering and transportation models

Use the basic methodology for the solution of linear programming problems

Understandthemathematicaltoolsthatareneededtosolveoptimizationproblemslike Transportationmodels andAssignmentmodels

Understandthereplacementmodelswithchangeinmoneyvalueconsideringwithtimeand without time

Modelasystemasaqueuingmodelandcomputeimportantperformancemeasures

Course Outcomes: at the end of the course student will be able to:

Understand the basics of Operations Research and solve linear programming problems using graphical, simplex method and dual simplex methods.

Use PERT and CPM techniques for project planning

Solve transportation and assignment problems using methods like MODI and Hungarian, including special cases like unbalanced and maximization problems

Understand how to plan for replacing equipment and use game theory to make decisions in competitive situations

Use sequencing methods to plan jobs on machines and study queuing models to understand and improve waiting line systems.

UNIT–I

Introduction: Definition and Scope of Operations Research. Linear Programming: Introduction, Formulation of linear programming problems, graphical method of solving LP problem, simplex method, maximization and minimization, Degeneracy in LPP, Unbounded and, Infeasible solutions

UNIT–II

Duality: Definition, Relationship between primal and dual solutions, Dual Simplex Method. Network Analysis in Project Planning: PERT&CPM–Cost Analysis and Crashing the network

UNIT–III

Transportation Models: Finding an initial feasible solution-North West corner method, least cost method, Vogel's Approximation method, Finding the optimal solution by MODI methods, Unbalanced Transportation problem. Assignment Problems: Hungarian method of Assignment problem, Maximization in Assignment problem, unbalanced problem, travelling salesman problems.

UNIT–IV

Replacement Models: Introduction, Replacement Policy for Items Whose Running Cost Increases with Time and Value of Money Remains Constant During a Period, Running Cost Increases with Time but Value of Money Changes with Constant Rate During a Period, Individual replacement policy, Group replacement policy.

Game Theory: Introduction, person zero sum games, Maximin -Minimax principle, Principle of Dominance, Solution for mixed strategy problems, Graphical method for 2xn and mx2 games.

UNIT–V

Sequencing Models: Introduction, General assumptions, processing n jobs through 2 machines, processing 'n' jobs through 3 machines, Processing 2 jobs through m machines. Queuing Theory: Introduction, single channel-Poisson arrivals with Exponential Service Time -Infinite Population and Service in Random Order, Generalization of Model(Birth and Death Process).

Suggested Readings:

1. Hamdy, A. Taha, “Operations Research-An Introduction”, Sixth Edition, Prentice Hall of India Pvt. Ltd.,1997
2. J.B.Gupta, “Utilization of Electric Power and Electric Traction”S.K.Kataria & Sons Publications, 2010
3. V.K.Kapoor, Operations Research,S.ChandPublishers,NewDelhi,2004
4. Hrvey M.Wagner, Principles of Operations Research, Second Edition, Prentice Hall of India Ltd., 1980.
R.Paneer Selvam, Operations Research, Second Edition, PHIL earning Pvt. Ltd., New Delhi, 2008.

Course Code	Course Title						Core/PE/OE
ES301EC	**BASIC ELECTRONICS**						**Core**
	Contact Hours per Week				**CIE**	**SEE**	**Credits**
Prerequisite	**L**	**T**	**D**	**P/D**			
BS202PH	**3**	**-**	**-**	**-**	**30**	**70**	**3**

Course Objectives: The course is taught with the objectives of enabling the student to:

1. Understand the characteristics of diodes and its applications.
2. Understand the design concepts of biasing of BJT and FET
3. Understand the design concepts of feedback amplifiers and oscillators
4. Study the design concepts of OPAmp.
5. Understand the concepts of Data Acquisition Systems and data converters

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

1. Study and analyze the rectifiers and regulator circuits.
2. Study and analyze the performance of BJTs, FETs on the basis of their operation And working.
3. Study&designoscillatorcircuits.
4. Study and analyze different Opamps and its applications.
5. Study and analyze different data acquisition systems

UNITI

PN Junction Diode: Characteristics, Half wav erectifier, Full wave rectifier, filters, ripple, regulation, TIF and efficiency, Zener diode and Zener diode regulators. CRT construction and CRO applications.

UNITII

Transistors: BJT construction and working, modes of operation, configurations of BJT (CB,CE,CC),small signal h-parameter model of CE, CE amplifier analysis. Construction and working of JFET, V-I characteristics of JFET.

UNITIII

Feedback concepts: Types of negative feedback– modification of gain, bandwidth, input and output impedances, applications.
Oscillators: RC Phase shift, Weinbridge, LC and crystal Oscillators (Qualitative treatment only).

UNITIV

Operational Amplifier: OP-AMP Blockdiagram, Ideal OP-AMP, DC and AC Characteristics, Inverting and Non-Inverting Amplifiers, Adder/Subtractor, Integrator, Differentiator, Precision rectifier , Schmit trigger and its applications.

UNITV

Data Acquisition Systems: Construction and Operation of transducers-Strain guage LVDT, Thermocouple, Instrumentation systems.
Data Converters: R-2R Ladder DAC, Successive approximation and Flash ADC.

Suggested TextBooks:
1.Robert BoylestadL. And Louis Nashelsky, Electronic Devices and Circuit Theory, PHI,2007.2st Edition, Prentice HallofIndia,2006.
2.Helfrick Dand David Cooper, Modern Electronic Instrumentation and MeasurementsTechniques,1st edition
3.Salivahanan, Suresh Kumar and Vallavaraj, Electronic Devices and Circuits,2nd Edition, TataMcGraw-Hill,2010.
Reference TextBooks:
1.David A. Bell, *Electronic Devices and Circuits*,5th ed., Oxford University Press, 2009
2.JBGupta, Electronic Devices and Circuits, S. K Kataria & sons,5thEdition,2012
3.The Art of Electronics, Horowitz,3rdEditionCambridgeUniversityPress
4.Electronic Devices and Circuits, A. P Godse, U. A Bakshi, Technical Publications

PC301CS	DATA STRUCTURES					
Prerequisites			**L**	**T**	**P**	**C**
			3	**-**	**-**	**3**
Evaluation	**CIE**	**30 Marks**	**SEE**		**70 Marks**	

Objectives:

1. To develop proficiency in the specification, representation, and implementation of abstract data types and data structures.
2. To discuss the linear and non-linear data structures and their applications
3. To introduce the creation, insertion and deletion operations on binary search trees and balanced binary search trees.
4. To introduce various internal sorting, searching techniques and their time complexities

Course Outcomes:

Essential Outcomes:

1. Understand the fundamentals of algorithm analysis and evaluate performance using time and space complexity along with asymptotic notations.
2. Implement linear data structures such as arrays, stacks, and queues, including their applications in expression evaluation, pattern matching, and sparse matrix handling.
3. Apply various types of linked lists (singly, circular, doubly) and perform dynamic memory management and operations such as polynomial manipulation and sparse matrix representation.
4. Design hierarchical data structures like trees and perform operations on BSTs and AVL trees for efficient searching, insertion, and deletion.
5. Apply graph traversal algorithms and internal sorting/searching techniques to solve computational problems using suitable data structures and efficient algorithms.

UNIT I

Algorithms: Introduction, Algorithm Specifications, Recursive Algorithms, Performance Analysis of an algorithm- Time and Space Complexity, Asymptotic Notations.

Arrays: Arrays ADT, Polynomials, Sparse matrices, Strings-ADT, Pattern Matching.

UNIT-II

Stacks and Queues: Stacks, Stacks using Arrays, Stacks using dynamic arrays, Evaluation of Expressions Evaluating Postfix Expression, Infix to Postfix. Queues: Queues ADT, operations, Circular Queues, Applications.

UNIT-III

Linked Lists: Singly Linked Lists and Chains, Linked Stacks and Queues, Polynomials, Operations for Circularly linked lists, Equivalence Classes, Sparse matrices, Doubly Linked Lists.

Hashing: Static Hashing, Hash Tables, Hash Functions, Overflow Handling, Theoretical Evaluation of Overflow Techniques.

UNIT - IV

Trees: Introduction, Binary Trees, Binary Tree Traversals, Heaps, Binary Search trees (BST): Definition, Searching an element, Insertion into a BST, Deletion from a BST.

Efficient Binary Search Trees: AVL Trees: Definition, Searching an element, Insertion into a AVL.

UNIT-V

Graphs: Graph Abstract Data Type, Elementary Graph operations (DFS and BFS), Minimum Cost Spanning Trees (Prim's and Kruskal's Algorithms).

Sorting and Searching: Insertion sort, Quick sort, Best computing time for Sorting, Merge sort, Heap sort, shell sort, Sorting on Several Keys, List and Table Sorts, Summary of Internal Sorting, Linear and Binary Search algorithms.

Suggested Books:

1. Horowitz E, Sahni S and Susan Anderson-Freed, Fundamentals of Data structures in C, 2nd Edition (Reprint 2024), Universities Press.

Reference Books:

1. Mark A Weiss, Data Structures and Algorithm Analysis In C, Second Edition (2002), Pearson.
2. Kushwaha D. S and Misra A.K, Data structures A Programming Approach with C, Second Edition (2014), PHI.
3. Gilberg R. F and Forouzan B. A, Data structures: A Pseudocode Approach with C, Second Edition (2007), Cengage Leaming
4. Tanenbaum A. M, Langsam Y. Augenstein M. J, Data Structures using C, Second Edition (2008), Pearson.
5. Thomas H. Cormen, Charles E. Leiserson, Ronald L Rivest, Clifford Stein, Introduction to Algorithms, Fourth Edition (2022), MIT Press
6. YedidyahLangsam, Moshe J. Augenstein, Aaron M. Tenenbaum, Data Structures Using C and C++, Second Edition (2009), PHI.

PC302CS	DISCRETE MATHEMATICS					
Prerequisites			**L**	**T**	**P**	**C**
			3	**-**	**-**	**3**
Evaluation	**CIE**	**30 Marks**	**SEE**		**70 Marks**	

Objectives:

1. To understand the concepts of propositional and predicate logic and their applications in theorem proving.
2. To explore the properties of relations, functions, and algebraic structures such as groups and lattices.
3. To develop skills in combinatorics and counting principles for solving real-world problems.
4. To solve recurrence relations using generating functions and other algebraic techniques.
5. To apply graph theory concepts to solve problems in networks, trees, and coloring using traversal and optimization algorithms.

Course Outcomes:

Essential Outcomes

1. Analyze logical arguments using truth tables, normal forms, quantifiers, and perform proof strategies such as contradiction and consistency.
2. Apply properties of binary relations and functions, including partial orders, equivalence relations, lattices, and algebraic structures.
3. Solve counting problems using combinations, permutations, binomial/multinomial theorems, and the principle of inclusion-exclusion.
4. Apply recurrence relations and generating functions to model and solve discrete problems
5. Analyse graphs using DFS and BFS and identify spanning trees and graph isomorphism.

UNIT-1

Mathematical Logic: Statements and notations, Connectives, Well-formed formulas, Truth Tables, tautology, equivalence implication, Normal forms, Quantifiers, universal quantifiers. Predicates: Predicative logic, Free & Bound variables, Rules of inference, Consistency, proof of contradiction, Automatic Theorem Proving

UNIT-II

Relations: Properties of Binary Relations, equivalence, transitive closure, compatibility and partial ordering relations, Lattices, Hasse diagram. Functions: Inverse Function Composition of functions, recursive Functions, Lattice and its Properties, Algebraic structures: Algebraic systems Examples and general properties, Semi groups and monads, groups sub groups' homomorphism, Isomorphism.

UNIT-III

Elementary Combinatorics: Basis of counting, Combinations & Permutations, with repetitions, Constrained repetitions, Binomial Coefficients, Binomial Multinomial theorems, the principles of Inclusion - Exclusion. Pigeon hole principles and its application.

UNIT-IV

Recurrence Relation: Generating Functions, Function of Sequences Calculating Coefficient of generating function, Recurrence relations, Solving recurrence relation by substitution and Generating. funds. Characteristics solution of in homogeneous Recurrence Relation.

UNIT-V

Graph Theory: Representation of Graph, DFS, BFS, Spanning Trees, planar Graphs. Graph Theory and Applications, Basic Concepts Isomorphism and Sub graphs, Multi graphs and Euler circuits, Hamiltonian graphs, Chromatic Numbers.

Suggested Readings:

1. Elements of Discrete Mathematics- A Computer Oriented Approach- CL Liu, D P Mohapatra. Third Edition, Tata McGrawHill.
2. Discrete Mathematics for Computer Scientists & Mathematicians, J.L. Mott, A. Kandel, T.P. Baker, PHI.
3. Discrete Mathematics and its Applications, Kenneth H. Rosen, Fifth Edition. TMH.
4. Discrete Mathematical Structures Theory and Application-Malik & Sen, Cengage.
5. Discrete Mathematics with Applications, Thomas Koshy, Elsevier
6. Logic and Discrete Mathematics, Grass Man & Trembley, Pearson Education.

PC303CS	OOP using JAVA					
Prerequisites			**L**	**T**	**P**	**C**
			3	**-**	**-**	**3**
Evaluation	**CIE**	**30 Marks**	**SEE**		**70 Marks**	

Objectives:

1. To understand fundamentals of object-oriented programming in Java which includes defining classes, invoking methods, difference between applet and application programs, using class libraries
2. To create Java application programs using sound OOP practices such as interfaces, exception handling, multi threading.
3. To understand fundamentals of object-oriented programming in Java which includes defining classes, invoking methods, difference between applet and application programs, using class libraries
4. Use Collection framework, AWT and event handling to solve real world problems.
5. Exploring Swing, and implementing Servlets.

Outcomes:

Course Outcomes:

1. CO1: Apply object-oriented programming principles such as encapsulation, inheritance, and polymorphism to solve basic problems using Java.
2. CO2: Develop and debug Java programs using classes, methods, constructors, arrays, exception handling, and interfaces.
3. CO3: Use multithreading and exception handling mechanisms to enhance performance and robustness of Java applications.
4. CO4: Employ Java Collection Framework and utility classes to manage and manipulate data efficiently.
5. CO5: Design and implement GUI-based applications using AWT and handle user events through event-driven programming.

UNIT-I

Object Oriented Programming: Principles, Benefits of Object Oriented Programming.

Introduction to Java: Java buzzwords, bytecode. Java Programming Fundamentals: Applet and Application program using simple java program, data types, variables, arrays, operators, expressions, control statements, type conversion and casting, concepts of classes, objects, constructors, methods, access control, this keyword, garbage collection, overloading methods and constructors, introducing access control, static, final, nested and inner classes, exploring string class, using command-linear arguments.

Inheritance: Inheritance concept, types of inheritance, Member access rules, use of super and final. Polymorphism - dynamic binding, method overriding, abstract classes and methods.

UNIT-II

Interfaces: Defining an interface, implementing interfaces, extending interface.

Packages: Defining, Creating and Accessing a Package, importing packages

Exception handling: Benefits of exception handling, classification, checked exceptions and unchecked exceptions, usage of try, catch, throw, throws and finally, rethrowing exceptions, built in exceptions, creating own exception sub classes

Multithreading: Java Thread Model, The Main Thread, creating a Thread, creating multiple threads, using is Alive() and join(), thread priorities, synchronization, inter thread communication, deadlock

UNIT-III

Collections: Overview of Java Collection frame work, commonly used Collection classes Array List, Linked List, Hash Set, Tree Set, Collection Interfaces Collection, List, Set. Accessing Collection via iterator, working with Map. Legacy classes and interfaces Vector, Hashtable, Stack, Dictionary, Enumeration interface.

Other Utility classes: String Tokenizer, Date, Calendar, Gregorian Calendar, ScannerJava Input/Output: exploring java.io, Java I/O classes and interfaces, File, Stream classes, byte stream, character stream, serialization.

UNIT-IV

GUI Programming with java: The AWT class hierarchy, MVC architecture. Applet Revisited: Basics, architecture and skeleton, simple applet program.
Event Handling: Delegation Event Model, Event Classes, Source of Events, Event Listener Interfaces. Handling mouse and keyboard events, Adapter classes.
Database Programming using JDBC: Introduction to JDBC, JDBC Drivers & Architecture, CURD operation Using JDBC, Connecting to non-conventional Databases.

UNIT V

Exploring Swing: JLabel, Imagelcon, JTextField, the Swing buttons, JTabbedpane, JScrollPane, JList, JComboBox.
Servlet: Life cycle, using tomcat, simple servlet, servlet API, javax.servlet package, reading servletparameters, javax.servlet.http package, handling HTTP requests and responses

Suggested Readings:

1. Herbert Scheldt, "The Complete Reference Java, 13th th Edition, Tata McGraw Hill, 2023.
2. James M Slack, Programming and Problem Solving with JAVA, Thomson Learning, 2002.
3. C Thomas Wu, An Introduction to Object Oriented Programming with Java 5th Edition, McGraw Hill Publishing, 2010.
4. H. M. Dietel and P. J. Dietel, Java How to Program, Sixth Edition, Pearson Education /PHI.

PC304CS	Logic and Switching Theory					
Prerequisites			**L**	**T**	**P**	**C**
			3	**-**	**-**	**3**
Evaluation	**CIE**	**30 Marks**	**SEE**		**70 Marks**	

Objectives

1. To impart knowledge on the fundamentals of digital computers, number systems, arithmetic operations, and information representation using digital codes.
2. To develop the ability to simplify Boolean expressions using algebraic and Karnaugh Map (K-Map) techniques and implement logic functions using universal gates.
3. To introduce combinational logic design, including standard building blocks such as adders, multiplexers, encoders, and decoders, and foster hierarchical design using hardware description languages.
4. To provide insights into sequential circuits through the design and analysis of flip-flops, latches, and synchronous state machines.
5. To explain the concepts of counters, registers, and symmetric networks and their synthesis in practical digital systems.

Course Outcomes:

At the end of the course, students will be able to:

1. Represent numerical and character data using various number systems and codes, and perform basic arithmetic operations using digital methods.
2. Simplify and implement Boolean expressions using K-Maps, tabulation methods, and universal gates such as NAND and NOR.
3. Design combinational logic circuits such as multiplexers, decoders, and arithmetic comparators using gate-level dataflow.
4. Distinguish between combinational and Sequential circuit operations.
5. Simulate and implement sequential circuits using latches and flip-flops.

UNIT-I

Digital Computers and Information: Information Representation, Computer Structure.
Number Systems: Binary Numbers, Octal and Hexadecimal Numbers, Number Ranges.
Arithmetic Operations: Conversion from Decimal to other bases.
Decimal Codes: BCD Addition and Subtraction. Alphanumeric Codes: ASCII Character Code, Parity Bit.
Binary Logic and Gates: Binary Logic, Logic Gates. Boolean Algebra: Basic Identifiers, Algebraic Manipulation, Complement of a Function.
Standard Forms: Minterms and Maxterms, Sum of Product and Products of Sums.

UNIT-II

Minimization of Boolean Functions: Introduction, the map methods (Karnaugh Map) and Minimal Functions and their Properties, the tabulation procedure (Quine McCluskey Method),
NAND and NOR Gates: Nand Circuits, Two-level Implementation, Multilevel NAND Circuits, NOR Circuits. Exclusive OR Gates: Odd Function, Parity Generation and Checking.

UNIT-III

Combination Logic Design: Combinational Circuits, Design Topics: Design Hierarchy, Top-Down design, Computer Aided Design, Hardware Description Languages, Logic Synthesis. Analysis Procedure: Derivation of Boolean Functions, Derivation of the Truth Table, Logic Simulation, Design Procedure, Decoders, Encoders, Multiplexers, Binary Adders, Binary Subtraction, Binary Multipliers,

UNIT-IV

Sequential Circuits: Sequential Circuit definitions, latches, Flip-Flops, Sequential circuit analysis, Sequential circuit design, design with SR Flip-Flop, D Flip-Flop, and T Flip-Flop, design with JK Flip-Flops, Registers and Counters: registers, Shift registers, Synchronous Binary counters, Ripple Counter.

UNIT-V

Design of Sequential Circuits: Basic Design Steps, Finite State Machine representation using Moore and Mealy State Models, State Minimization, Design of FSM for Sequence generation and Algorithmic State Machine Charts, Symmetrical functions and their representations

Suggested Reading:

1. M. Moris Mano, Charles R. Kime, Logic and Computer Design Fundamentals, 5th edition, Pearson Education Asia, 2001.
2. Zvi Kohavi, Switching and Finite Automata Theory, 2nd edition, Tata McGraw Hill, 1995.
3. Charles H. Roth, Jr Fundamentals of Logic Design, 5th edition, Thomson, Brook, Cole, 2005.
4. Ref: AICTE e-Kumbha portal (Digital Electronics and Systems by Dr. Abhisek Bhatt)

PC351CS	**Data Structures Lab**					
Prerequisites			**L**	**T**	**P**	**C**
			-	**-**	**2**	**1**
Evaluation	**CIE**	**25 Marks**	**SEE**		**50 Marks**	

Objectives:

1. To develop skills to design and analyse simple linear and nonlinear data structures.
2. To gain programming skills to implement sorting and searching algorithms
3. To Strengthen the ability to identify and apply the suitable data structures for the given real world problem
4. To Gain knowledge in practical applications of data structures

Outcomes:

After completing this course, the student will be able to:

1. Implement linear data structures such as arrays, stacks, queues, and their variations to perform basic operations like insertion, deletion, and traversal.
2. Apply linked lists (singly, doubly, and circular) for dynamic memory management and use them to implement linear data structures such as stacks and queues.
3. Use standard searching (linear and binary) and sorting techniques (selection, insertion, merge, quick, heap) to process and organize data efficiently.
4. Construct tree structures such as binary trees, binary search trees, and AVL trees, and apply recursive traversal techniques.
5. Apply graph traversal (DFS and BFS), hashing techniques, and minimum spanning tree algorithms (Prim's and Kruskal's) for solving computing problems.

List of Experiments:

1. Write a program to represent arrays for the following:
    a. To check whether the given matrix is sparse or not and display triplet representation.
    b. To add two given polynomials.
2. Write a program to apply arrays to perform operations on the following linear data structures:
    a. Stacks
    b. Queues
    c. Circular Queue
3. Write a program to implement following stack applications:
    a. String Reversal.
    b. Infix to Postfix Conversion.
    c. Postfix Expression Evaluation.
    d. Balanced Parenthesis
4. Write a program to implement the operations of Singly Linked List
5. Write a program to implement the operations of Doubly Linked List.
6. Write a program to implement the operations of Circular Linked List.
7. Write a program to apply linked list to perform operations on the following data structures:
    a. Stacks
    b. Queues
8. Write a program to search for a an element given in an array using following search techniques:
    a. Linear search
    b. Binary search
9. Write a program to build a hash table using linear probing and search for a given element.

10. Write a program to construct a Binary Tree and implement display in-order, pre-order and post-order traversal.
11. Write a program to construct a Binary Search Tree and implement insertion, deletion and search operations on it.
12. Write a program to construct a AVL tree and implement insertion, deletion and search operations on it.
13. Write a program to connect a graph and traverse the graph using DFS and BFS.
14. Write a program to design a minimum spanning tree from a given graph using Prims and Kruskals Algorithm.
15. Write a program to sort given set of elements using following sorting techniques:
    a. Selection sort
    b. Insertion sort
    c. Merge sort
    d. Quick sort
    e. Heap sort

Course Code	Course Title						Core/PE/OE
ES351EC	**BASIC ELECTRONICS LAB**						**Core**
	Contact Hours per Week				**CIE**	**SEE**	**Credits**
Prerequisite	**L**	**T**	**D**	**P**			
ES301EC				**2**	**25**	**50**	**1**
Course Objectives: The course is taught with the objectives of enabling the student to:							
1.To understand the characteristics of diodes and transistor configurations.							
2.To understand the design concepts of biasing of BJT and FET							
3.To understand the design concepts of feedback amplifiers and oscillators							
4.To study the design concepts of OPAmp and data converters							
Course Outcomes: On completion of this course, the student will be able to:							
1.Ability to design diode circuits & understand the application of Zener diode.							
2.Ability to analyze characteristics of BJTs & FETS.							
3.Ability to understand the different oscillator circuits.							
4.Ability to understand operation of HWR & FWR circuits with & without filters.							
5.AbilitytomdesignAnalog-to-Digitalconverters&Digital-to-Analogconverters.							

LISTOF EXPRIMNETS
1.CRO-Applications, Measurements of R. Land C using LCR meter, Colour code Method and soldering practice.
2.Characteristics of Semiconductors diode Ge. Si and Zener.
3.Static Characteristics of BJT-Common Emitter.
4.Static Characteristics of BIT-Common Base.
5.Static Characteristics of FET.
6.RC-Phase Shift Oscillator.
7.Hartleyand Colpitts Oscillators.
8.Common Emitter Amplifier.
9.A stable Multivibrator.
10.Full-wave rectifier with and without filters using BJT.
11.Operational Amplifier Applications.
12.Strain Gauge Measurement.
13.Analog-to-DigitalandDigitaltoAnalog Converters.
Suggested TextBooks:
1.PaulB. Zbar, Albert P. Malvino, Michael A.Miller, *Basic Electronics, A Text-Lab Manual*, 7th ed., Mc Graw Hill Education, 2001.
2.David Bell, *Fundamentals of electronic devices and circuits LabManual*,5th ed., Oxford universitypress,2009.
3.R.C.Jaeger & T.N.Blalock, *Micro Electronic circuit design*,4th ed.,McGrawHill Higher Education,2011.

PC352CS	**OOP using JAVA Lab**					
Prerequisites			**L**	**T**	**P**	**C**
			-	**-**	**2**	**1**
Evaluation	**CIE**	**25 Marks**	**SEE**		**50 Marks**	

Objectives:

The objectives of the course are to impart knowledge of:
1. To build software development skills using java programming for real world applications.
2. To implement frontend and backend of an application
3. To implement classical problems using java programming.

Outcomes:
After the completion of the course, the student will be able to:

1. Develop Java applications using the concepts of Inheritance, interfaces, packages, access control specifiers.
2. Implement the concepts of Exception Handling in java Applications.
3. Read and write data using different Java I/O streams.
4. Create graphical user interfaces and Applets by applying the knowledge of Event Handling.
5. Create robust applications using Java standard class libraries and retrieve data from a database with JDBC.
6. Ability to solve real-world problems by designing user friendly GUI with befitting backend through the APIs of Java.

List of Experiments:

1. Write a Java program to illustrate the concept of class with method overloading
2. Write a Java Program that reads a line of integers, and then displays each integer, and the sum of all the integers (Use String Tokenizer class of java.util)
3. Write a Java program to illustrate the concept of Single level and Multi level Inheritance.
4. Write a Java program to demonstrate the Interfaces & Abstract Classes.
5. Write a Java program to implement the concept of exception handling.
6. Write a Java program to illustrate the concept of threading using Thread Class and runnable Interface.
7. Write a Java program to illustrate the concept of Thread synchronization.
8. Write a Java program that correctly implements producer consumer problem using the concept of inter thread communication.
9. Write a Java program to illustrate collection classes like Array List, LinkedList, Treemap and Hash map.
10. Write a Java program to illustrate Legacy classes like Vector, Hash table, Dictionary & Enumeration interface.
11. Write a Java program to implement iteration over Collection using Iterator interface and List Iterator interface.
12. Write a Java program that reads a file name from the user, and then displays information about whether the file exists, whether the file is readable, whether the file is writable, the type of file and the length of the file in bytes.
13. Write a Java program to illustrate the concept of I/O Streams

14. Write a Java program to implement serialization concept
15. Write a Java applet program to implement Colour and Graphics class
16. Write a Java applet program for handling mouse & key events

17. Write a Java applet program to implement Adapter classes
18. Write a Java program that works as a simple calculator. Use a grid layout to arrange buttons for the digits and for the +, -, *, % operations. Add a text field to display the result.
19. Write an example for JDBC prepared statement with ResultSet
20. Write a Java Program to get primary key value (auto-generated keys) from inserted queries using JDBC
21. Write a Java Program to create a simple JList
22. Write a Java Program to create a simple checkbox using JCheckBox
23. Write a Java Program to create a checkbox and Item Listener toit.
24. 1. Write Servlet application to print current date &time
2. Html & Servlet Communication
3. Auto refresh a page
4. Demonstrate session tracking
5. Select record from database
6. Application for login page
7. Insert record into database
8. Count the visits on webpage
9. Insert teacher record in Database.

SEMESTER-IV

S. No.	Course Code	Course Title	Scheme of Instruction				Scheme of Examination			Credits
			L	T	D/P	Contact Hrs/Wk	CIE	SEE	Duration in Hrs	
Theory Courses										
1	PC 401 CS	Automata Languages & Computation	3	-	-	3	30	70	3	3
2	HS 406 CM	Managerial Economics and Accounting	3	-	-	3	30	70	3	3
3	BS 207 MT	Probability & Statistics	3	1	-	4	30	70	3	4
4	PC 402 CS	Operating Systems	3	-	-	3	30	70	3	3
5	PC 403 CS	Computer Organization	3	-	-	3	30	70	3	3
6	ES 401 EC	Signals and Systems	3	-	-	3	30	70	3	3
7	PC 404 CS	Database Management Systems	3	-	-	3	30	70	3	3
Practical / Laboratory Courses										
8	PC 451 CS	Computer Organization Lab	-	-	2	2	25	50	3	1
9	PC 452 CS	Operating Systems Lab	-	-	2	2	25	50	3	1
10	PC 453 CS	Database Management Systems Lab	-	-	2	2	25	50	3	1
Total			**21**	**1**	**6**	**28**	**285**	**640**	**30**	**25**

PC401CS	**Automata Languages and Computation**					
Prerequisites			**L**	**T**	**P**	**C**
			3	**-**	**-**	**3**
Evaluation	**CIE**	**30 Marks**	**SEE**		**70 Marks**	

Objectives:

1. To understand the fundamental concepts of Finite Automata, Regular Expressions, and their applications.
2. To construct and analyze Context-Free Grammars (CFGs) and Pushdown Automata (PDAs) for language recognition.
3. To examine properties of Context-Free Languages including normal forms, pumping lemma, and closure properties.
4. To study Turing Machines and their capabilities in solving computational problems.
5. To explore and evaluate the decidability and undecidability aspects of languages using theoretical models.

Course Outcomes:

1. Construct deterministic anautomata and regular expressions for recognizing regular languages.
2. Develop context-free grammars and design PDAs for context-free languages and analyze their equivalence.
3. Convert CFGs into Chomsky and Greibach Normal Forms and apply the CFL Pumping Lemma.
4. Demonstrate the design and functioning of Turing Machines and explain various machine modifications.
5. Distinguish and analyze recursive, recursively enumerable languages, and undecidability problems.

UNIT I

Introduction: Finite state automata, Non-deterministic finite state automata, FA with E-transitions, Regular expressions, Applications of FA, Properties of regular sets, Pumping Lemma, Closure properties, Myhill-Nerode Theorem, Minimization of FA.

UNIT-II

Context Free Grammars and Languages: Derivations, Parse-trees, Ambiguity in Grammars and Languages. Pushdown Automata-Definitions, The languages of PDA, Equivalence of PDAs and CFGs, Deterministic Pushdown Automata.

UNIT-III

Properties of CFLs: Normal forms for CFGs, Pumping Lemma, Closure properties, Deterministic Context Free Languages, Decision properties.

UNIT-IV

Turing Machines: Introduction, Computational Languages and Functions, Techniques for construction of Turing machines. Modifications of TM, TM as enumerator, Restricted TM.

UNIT-V

Undecidability: Recursive and Recursively enumerable languages, UTM and undecidable problem, Rice Theorem, Post's correspondence problem. Chomsky's Hierarchy-Regular grammars, Unrestricted grammar, CSL, Relationship between classes of languages.

Textbooks:

1. John E. Hopcroft, Rajeev Motwani, and Jeffrey D. Ullman *Introduction to Automata Theory, Languages, and Computation* 3rd Edition, Pearson Education, 2021
2. Peter Linz *An Introduction to Formal Languages and Automata*, 6th Edition, Jones & Bartlett Learning, 2022

Reference Books:

1. Michael Sipser *Introduction to the Theory of Computation*, 3rd Edition, Cengage Learning, 2018.
2. K.L.P. Mishra and N. Chandrasekaran, *Theory of Computer Science: Automata, Languages and Computation,* 4th Edition, PHI Learning, 2020
   Daniel I.A. Cohen *Introduction to Computer Theory*, 2nd Edition, Wiley India, latest reprint 2021.

Online Resources (for supplementary learning):

NPTEL Courses by Prof. Kamala Krithivasan (IIT Madras) – *Theory of Computation.*

MANAGERIAL ECONOMICS AND ACCOUNTANCY

HS406CM

Instruction: 3 periods per week — Duration of SEE:3 hours

CIE:30marks — SEE:70 marks Credits:3

Course Objectives:

To understand responsibilities of a manager of a business undertaking.

To analyze various factors influencing demand elasticity

To Forecast & compute the future sales level.

To understand the features, steps, merits, uses & limitations of Pay Back, ARR, NPV, PI & IRR methods of Capital Budgeting

To understand the principles of accounting and prepare Journal, Ledger, Trial Balance, Manufacturing A/c, Trading A/c., Profit & LossA/c. and Balance Sheet of an enterprise.

Course Outcomes:

Upon successful completion of this course, the student will be able to

Understand the responsibilities of a manager of a business undertaking

Able to Forecast & compute the future sales level

Outline the features, steps, merits, uses & limitations of Pay Back, ARR, NPV, PI& IRR methods of Capital Budgeting

Assess various factors influencing demand elasticity and determine Break Even Point (BEP) of an enterprise.

Understands the principles of accounting and prepare Journal, Ledger, Trial Balance, Manufacturing A/c, Trading A/c., Profit & LossA/c. and Balance Sheet of an enterprise.

UNIT-I

Introduction to economics and its evolution: Managerial Economics its Scope, Importance and relation to other sciences, its usefulness to engineers-Basic concepts of Managerial Economics.

UNIT-II

Demands: Analysis-concept of demand, determinants, law of demand, its assumptions, elasticity of demand, price, income and cross elasticity, demand forecasting-markets competitive structure, price- output determination under perfect competition and Monopoly.

UNIT-III

Theory of Production: Firm and industry-production function-input-output relations-laws of returns- internal and external economics of scale. Cost analysis-Cost concepts-fixed and variable costs-explicitly and implicitly costs-out pocket of costs and imputed costs-opportunity cost-cost output relation- ship-break even analysis.

UNIT-IV

Capital management: Significance, determinates and estimation of fixed and working capital requirements, sources of capital. Introduction to capital budgeting, methods of payback and discounted cash flow methods with problems.

UNIT-V

Book-keeping: Principles and significance of double entry book keeping, journal, subsidiary books,

ledger accounts, trial balance concepts and preparation of final accounts with simple adjustments- analysis and interpretation of financial statements through ratios.

SuggestedReadings:

1. Varshney, R. L., and K. L. Maheshwari. Managerial Economics. Sultan Chand& Sons. 2010.

2. Eugene F. Brigham, James L. Pappas, Managerial economics, DrydenPress,1979

3. Grawal T.S.S C Gupta, Introduction to Accountancy, S Chand Publications,1978

4. I. M. Panday I.M., Financial Management, Vikas Publishing House Pvt Limited, 11th Ed. 2015

5. S K Maheshwari S N Maheshwari, An Introduction to Accountancy, 8th Ed.Vikas Publishing House Pvt Limited, 2006

PC402CS	OPERATING SYSTEMS					
Prerequisites			**L**	**T**	**P**	**C**
			3	**-**	**-**	**3**
Evaluation	**CIE**	**30 Marks**	**SEE**		**70 Marks**	

Objectives:

1. To learn the fundamentals of Operating Systems.
2. To learn the mechanisms of OS to handle processes and threads and their communication.
3. To learn the mechanisms involved in memory management in contemporary OS.
4. To gain knowledge on distributed operating system concepts that includes architecture, Mutual exclusion algorithms, deadlock detection.
5. To know the components and management aspects of concurrency management.

Outcomes:

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

CO1. Describe the evolution, types, and structure of operating systems including system calls and Virtual Machine.

CO2. Identify and explain the lifecycle of processes and threads along with scheduling techniques

CO3. Apply various synchronization mechanisms to resolve concurrency problems in operating systems and Deadlock conditions

CO4. Demonstrate different memory management schemes including paging and virtual memory concepts.

CO5. Analyze and compare different file systems and I/O management techniques.

UNIT-1

Introduction: Concept of Operating Systems, Generations of Operating systems, Types of Operating Systems, OS Services, System Calls, Structure of an OS Layered, Monolithic, Microkernel Operating Systems, Concept of Virtual Machine.

UNIT-II

Processes: Definition, Process Relationship, Different states of a Process, Process State transitions, Process Control Block (PCB), Context switching Thread: Definition, Various states, Benefits of threads, Types of threads, Concept of multithreads, Process Scheduling: Foundation and Scheduling objectives, Types of Schedulers, Scheduling Criteria, Scheduling algorithms, multiprocessor scheduling.

UNIT-III

Process Synchronization: Inter-process Communication: Critical Section, Race Conditions, Mutual Exclusion, Peterson's Solution, classical problems of synchronization: The Bounded buffer problem, Producer Consumer Problem, reader's & writer problem, Dinning philosopher's problem. Semaphores, Event Counters, Monitors, Message Passing, Deadlocks: Definition, Necessary and sufficient conditions for Deadlock, Methods for Handling: Deadlocks: Deadlock prevention, Deadlock Avoidance: Banker's algorithm, Deadlock detection and Recovery.

UNIT-IV

Memory Management: Basic concept, Logical and Physical address map, Memory allocation: Contiguous Memory allocation, fragmentation, and Compaction; Paging: Principle of operation - Page allocation - Hardware support for paging, structure of page table, Protection and sharing, Disadvantages of paging.

Virtual Memory: Basics of Virtual Memory - Hardware and control structures - Locality of reference, Page fault, Working Set, Dirty page/Dirty bit - Demand paging, Page Replacement algorithms, Trashing.

UNIT - V

I/O Hardware: I/O devices, Device controllers, Direct memory access Principles of I/O Software: Goals of Interrupt handlers, Device drivers, Device independent I/O software,

File Management: Concept of File, Access methods, File types, File operation, Directory structure, File System structure, Allocation methods, Free-space management, directory implementation, efficiency, and performance.

Secondary-Storage Structure: Disk structure, Disk scheduling algorithms, Disk Management, RAID structure.

Suggested Readings:

1. Avi Silberschatz, Peter Galvin, Greg Gagne, Operating System Concepts Essentials, 9th Edition, Wiley Asia Student Edition, 2017.
2. William Stallings, Operating Systems: Internals and Design Principles, 5th Edition, Prentice Hall of India, 2016.
3. Maurice Bach, Design of the Unix Operating Systems, 8th Edition, Prentice-Hall of India, 2009.
4. Daniel P. Bovet, Marco Cesati, Understanding the Linux Kernel, 3rd Edition,, O'Reilly and Associates.

PC403CS	Computer Organization					
Prerequisites			**L**	**T**	**P**	**C**
			3	**-**	**-**	**3**
Evaluation	**CIE**	**30 Marks**	**SEE**		**70 Marks**	

Objectives:

1. To introduce the fundamental concepts of computer architecture, including functional units, bus structures, and input/output organization.
2. To explain the organization and functioning of memory systems, including RAM, ROM, cache, and virtual memory, and their impact on system performance.
3. To provide a comprehensive understanding of the 8086 microprocessor architecture, instruction set, and assembly-level programming.
4. To develop the ability to interface various I/O devices with the 8086 microprocessor and understand the use of interrupts and peripheral controllers.
5. To expose students to real-time applications of microprocessor-based systems in embedded and IoT development using programmable peripheral interfaces.

Course Outcomes

1. Describe the functional units of a computer, types of computers, and various I/O mechanisms including DMA and interrupt-driven I/O.
2. Explain the principles of memory organization, including RAM, ROM, cache, and virtual memory concepts.
3. Analyze the architecture and internal operations of the 8086 microprocessor and interpret instruction sets and addressing modes.
4. Apply assembly-level programming techniques for basic problem-solving using 8086 instructions and subroutine handling.
5. Demonstrate how to interface output displays and input devices (e.g., keyboards) with microprocessors using peripheral support chips.

UNIT-1

Basic Structure of Computers: Computer Types, Functional Units, Basic Operational Concepts, Bus Structures, Performance, Multiprocessors and Multicomputers, Historical perspective. Input/output Organization: Accessing I/O devices, Interrupts, Processor examples, Direct memory access, parallel interface and serial interface.

UNIT-II

The Memory System: Basic concepts, Semiconductor RAM memories, Read-Only memories, Speed, Size and Cost, Cache memories, Performance considerations, Virtual Memories, Memory management requirements, Secondary Storage.

UNIT-III

8086 Architecture: Introduction to microprocessors and microcontrollers, 8086 Processor Architecture, Internal operations, Instructions and timings. Programming the 8086 Introduction to 8086 instructions, Addressing modes and Programming techniques with Additional instructions.

UNIT-IV

Stacks and subroutines, interfacing peripherals Basic interfacing concepts, interfacing output displays, Interfacing input keyboards. Interrupts 8086 Interrupts, Programmable peripheral interface (Intel 8255A),

UNIT-V

8251, 8253/8254, 8257 8259, real world case study of 8086 with peripheral devices, Application of further to embedded systems development and IOT development

Suggested Readings:

1. Carl Hamacher, Zvonko Vranesic, SafwatZaky, Computer Organization, 5th Edition, McGraw Hill, 2002.
2. Fundamentals Of Microprocessors and Microcontrollers 8/E PB Paperback – 1 January 2012 by Ram B Dhanpat Rai Publishers
3. Microprocessors and Interfacing Paperback – 23 May 2012, by M. Saravanan N. Senthil Kumar , S. Jeevananthan r), S.K.Shah OXFORD PUBLICATIONS

References

1. Embedded / Real-Time Systems: Concepts, Design and Programming Black Book, New ed Paperback – 1 January 2003 by Dr. K.V.K Prasad , Dreamtech Publishers
2. VLSI Design Black Book by Dr. K.V.K.K. Prasad , Dr. Kattula Shyamala Dreamtech Publishers
3. 21 IoT Experiments: Learn IoT, the Programmer's way Paperback – 1 January 2017
by Yashavant Kanetkar , Shrirang Korde BPB PUBLICATIONS

PC404CS	**Database Management Systems**					
Prerequisites			**L**	**T**	**P**	**C**
			3	**-**	**-**	**3**
Evaluation	**CIE**	**30 Marks**	**SEE**		**70 Marks**	

Course Objectives:

1. Understand core database concepts, including file organization methods, database architecture, schema levels, and relational algebra operations.
2. Model real-world scenarios using E-R diagrams and convert them into normalized relational schemas following Codd's rules.
3. Construct advanced SQL queries using joins, subqueries, set operations, and aggregation to retrieve and manipulate data effectively.
4. Apply normalization techniques to refine database design, eliminate redundancy, and maintain data consistency and integrity.
5. Develop PL/SQL programs using control structures, triggers, procedures, and indexing to enhance database functionality and performance.

Course Outcomes:

CO1: Understand database systems fundamentals for file organization, database environment, schemas, and relational algebra operations.
CO2: Design E-R models and convert them into normalized relational schemas using E.F. Codd's rules and solve complex SQL queries.
CO3: Analyze the need for and apply normalization techniques to remove data redundancy and ensure data integrity.
CO4:. Develop PL/SQL programs using control structures, cursors, triggers, procedures, and functions; apply indexing techniques to optimize database performance and enforce complex integrity constraints
CO5: Understand the concepts of transactions and develop an overview of diverse databases.

UNIT – I
Introduction to Database: File System Organization: Sequential - Pointer - Indexed – Direct, Purpose of Database System, Database Characteristics, Users of Database System, Advantages of DBMS Approach, Schemas and Instances - Three Schema Architecture and Data Independence, Database System Environment. Relational Algebra – Selection and projection, Renaming, Set operations Joins, Expressive Power of Algebra and calculus
UNIT – II
Logical Database Design: Design of Relational Database – E.F. Codd's Rule - Entity-Relationship model, Weak Entity, Strong Entity, Attributes, Extended ER Diagrams. Structured Query Language: Structured Query Language (SQL): Form of Basic SQL Query, Examples of Basic SQL Queries, Introduction to Sub-Queries and Nested Queries, Set operations in SQL , Comparison Operators and Single row & Multi-row operators,, Aggregative Operators, NULL value, , Disallowing NULL values, Logical connectivity's – AND, OR and NOT, Logical Operators - ALL, ANY, IN, BETWEEN, EXISTS, LIKE, NOT, SOME , Joins – LEFT, RIGHT, OUTER, NATURAL
UNIT – III

Integrity constraints: Integrity Constraint Over relations – Enforcing Integrity constraints – Querying relational data – Logical database Design – Introduction to Views – Destroying /altering Tables and Views, User-level and System-level privileges. Normalization: Introduction to Lossless and Lossy decomposition and functional dependencies, First, Second, and third normal forms – dependency preservation, Boyee/Codd normal form, and other normal forms examples. (4th and 5th Normal forms).
UNIT – IV
Introduction to PL/SQL: Control Structures, Cursors, SQL Triggers, Procedures, and Functions, Complex Integrity Constraints in SQL Triggers and Active Data bases. Indexing:Types of Single Level Ordered Indexes - Multilevel Indexes - Dynamic Multilevel Indexes
UNIT – V
Transaction Processing and Concurrency Control: Transaction Concepts - ACID Properties – Transaction States - Concurrency Control Problems - Serializability - Recoverability - Pessimistic and Optimistic Concurrency Control Schemes. Advanced Topics: Overview: Parallel Database - Multimedia Database - Mobile Database - Web Database - Multidimensional Database. Data Warehouse - OLTP Vs OLAP - NoSQL Database.

Suggested Reading:

1. Abraham Silberschatz, Henry F Korth, S. Sudarshan, Database System Concepts, Sixth Edition, McGrah-Hill International Edition, 2010.
2. Ramakrishnan, Gehrke, Database Management Systems, Third Edition, McGrah-Hill International Edition, 2003.
3. Elmasri Navathe, Somayajulu, Fundamentals of Database System, Fourth Edition, Pearson Education, 2006.
4. Patric O'Neil, Elizabeth O'Neil, Database--principles, programming, and performance, Morgan Kaufmann Publishers, 2001.

Reference Book from ekumb-AICTE :
1. Dr. Madhu Bala Myneni , Introduction to DBMS: Theory & Practicals, Language- English & Telugu url link: https://ekumbh.aicte-india.org/allbook.php#

PC451CS	**Operating Systems Lab**					
Prerequisites			**L**	**T**	**P**	**C**
			-	**-**	**2**	**1**
Evaluation	**CIE**	**25 Marks**	**SEE**		**50 Marks**	

Objectives:
1. Learn different types of CPU scheduling algorithms.
2. Demonstrate the usage of semaphores for solving synchronization problem.
3. Understand memory management techniques and different types of fragmentation.
4. That occur in them and various page replacement policies.
5. Understand Banker's algorithm used for deadlock avoidance.
6. Learn various disk scheduling algorithms.

Outcomes:
Student will be able to
Evaluate the performance of different types of CPU scheduling algorithms.
Implement producer-consumer problem, reader-writers problem, Dining philosopher's problem.
Simulate Banker's algorithm for deadlock avoidance.
Implement paging replacement and disk scheduling techniques.
Use different system calls for writing application programs.

1. CASE STUDY
Perform a case study by installing and exploring various types of operating systems on a physical or logical (virtual) machine.

II. List of Experiments (preferred programming language is C)
1. Write a C programs to implement UNIX system calls and file management
2. Write C programs to demonstrate various process related concepts.
3. Write C programs to demonstrate various thread related concepts.
4. Write C programs to simulate CPU scheduling algorithms: FCFS, SJF, Round Robin
5. Write C programs to simulate Intra & Inter-Process Communication (IPC) techniques: Pipes, Messages Queues, Shared Memory.
6. Write C programs to simulate solutions to Classical Process Synchronization
Problems: Dining Philosophers. Producer-Consumer, Readers-Writers
7. Write a C program to simulate Bankers Algorithm for Deadlock Avoidance.
8. Write C programs to simulate Page Replacement Algorithms: FIFO, LRU
9. SSTF Write C programs to simulate implementation of Disk Scheduling Algorithms: FCFS,

Software Required:
StarUML/Umbrello, NetBeans/Eclipse IDE, XAMPP/MEAN stack, JUnit, JMeter, Selenium, Bugzilla.

PC452CS	**Computer Organization Lab**					
Prerequisites			**L**	**T**	**P**	**C**
			-	**-**	**2**	**1**
Evaluation	**CIE**	**25 Marks**	**SEE**		**50 Marks**	

Objectives:

The objectives of the course are to impart knowledge of:

1. To become familiar with the architecture and Instruction set of Intel 8086 microprocessor.
2. To provide practical hands on experience with Assembly Language Programming.
3. To familiarize the students with interfacing of various peripheral devices with 8086 microprocessors using 8086 kits or MASM ASSEMBLER .

Outcomes:

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

1. Interpret the principles of Assembly Language Programming, instruction set in developing microprocessor based applications.
2. Develop Applications such as: 16-bit Addition, Multiplication, Division, array operations, swapping, negative and positive numbers.
3. Build interfaces of Input-output and other units like stepper motor with8086. Analyse the function of traffic light controller.

List of Programs:

1. Tutorials on 8086Programming.
2. Interfacing and programming of 8255. (E.g. traffic light controller).
3. Stepper motor interface.
4. A/D and D/A converter interface.
5. Display interface
6. real world case study of 8086 with peripheral devices,
7. Application of further to embedded systems development and IOT development

PC453CS	Database Management Systems Lab					
Prerequisites			**L**	**T**	**P**	**C**
			-	**-**	**2**	**1**
Evaluation	**CIE**	**25 Marks**	**SEE**		**50 Marks**	

Objectives:
The objectives of the course are to impart knowledge of:
1. To practice various DDL commands in SQL
2. To write simple and Complex queries in SQL
3. To familiarize PL/SQL

Outcomes:
After the completion of the course, the student will be able to:
1. Design and implement a database schema for a given problem
2. Develop the query statements with the help of structured query language.
3. Populate and query a database using SQL and PL/SQL
4. Develop multi-user database application

Design GUI using forms and implement database connectivity:

List of Programs
1. Creation of database (exercising the commands for creation)
2. Simple condition query creation using SQL Plus
3. Complex condition query creation using SQL Plus
4. Usage of Triggers and Stored Procedures.
5. Creation of Forms for student Information, library information, Pay roll etc.
6. Writing PL/SQL procedures for data validation
7. Generation using SQL reports
8. Creating Password and Security features for applications.
9. Usage of File locking table locking, facilities in applications.
10. Creation of small full pledged database application spreading over to 3sessions.

Suggested Readings:
1. Nilesh Shah, Database System Using Oracle, PHI, 2007.
2. Rick F Vander Lans, Introduction to SQL, Fourth edition, Pearson Education, 2007.
3. Benjamin Rosenzweig, Elena Silvestrova, Oracle PL/SQL by Example, Third edition, Pearson Education, 2004.
4. Albert Lulushi, Oracle Forms Developer's Handbook, Pearson Education, 2006.