Study Scheme and Syllabus for Five-Year Integrated M. Tech. Programme in Computer Science and Engineering

SEMESTER-I

Study Scheme and Syllabus for Five-Year Integrated M. Tech. Programme in Computer Science and Engineering

Title of the course : **Elements of Computer Engineering**

Subject code : **ESCS-401**

Weekly load : **2 Hours** **LTP 2-0-0**

Credits : **2**

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

CO1	Get acquainted with basics of computer system along with its various components
CO2	Know about various operating systems and memory
CO3	Study the C programming basics and learn the concept of operators
CO4	Understand the concept of decision statements and loops
CO5	Learn the use of functions, pointers, arrays, structures, union etc. for modular programming

	CO/PO Mapping: (Strong(3)/Medium(2)/Weak(1) indicates strength of correlation):													
COs	Program Outcomes (PO's)/ Program Specific Outcomes (PSO's)													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	1	3	1	3	0	0	0	3	1	3	2	3	3
CO2	3	1	3	3	3	0	0	0	3	2	3	2	3	3
CO3	3	2	3	1	3	3	0	0	2	2	3	2	3	3
CO4	3	2	3	3	3	3	0	0	3	3	3	3	3	3
CO5	3	3	3	3	3	3	0	0	3	3	3	3	3	3

Theory

Unit	Main Topics	Course outlines	Lecture(s)
Unit-1	1. Fundamentals of Computer Systems	Introduction and characteristics of computer systems, Generations, Classifications, Applications, Central Processing Unit, Memory, I/O devices, Introduction to Operating System and its types, Algorithm, Flowchart.	04
	2. C Programming Basics	Basic program construction, Structure of a C program, Compilation process, Preprocessor directives, Comments, Data types, Type conversions, Operators - Arithmetic, Relational, Logical, Conditional, Increment/decrement, Library functions, Header files.	04
	3. Loops and Decision Statements	*for* loop, *while* loop, *do* loop, Various forms of *if* statement,*switch* statement, *break* statement, *continue* statement, *goto* statement.	04
	4. Functions	Defining functions, Passing arguments to functions, Returning values from functions, Reference arguments, Storage classes.	04
Unit-2	5. Pointers	Pointers, Pointers to pointers, Declaring and initializing pointers, Pointer expressions, Pointers and arrays, Pointers and strings.	04
	6. Arrays	Arrays and strings, Declaring an array, Initializing arrays, Accessing the array elements, Working with multidimensional arrays, Declaring and initializing string variables, String handling functions.	04
	7. Structures	Declaring and initializing a structure, Accessing the	04

	and Unions	members of a structure, Nested structures, Array of structures, Using structures in functions, Pointers and structures, Declaring and initializing a union.	
	8. Files	Reading and writing to text and binary files, Character I/O, String I/O, File pointers, Error handling, Redirection, Command line arguments.	04

Total=32

Recommended Books:

1. Raja Raman V., Fundamentals of Computers, PHI.
2. Kernighan Brian W. and Ritchie, Dennis M., The C Programming language, Dorling Kingsley.
3. Balagurusamy E., Programming in ANSI C, TMH Publications

Study Scheme and Syllabus for Five-Year Integrated M. Tech. Programme in Computer Science and Engineering

Title of the course : **Elements of Computer Engineering Lab**

Subject Code : **ESCS-402**

Weekly load : **4 Hours** **LTP 0-0-4**

Credits : **2**

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

CO1	Solve basic mathematical problems using programming
CO2	Demonstrate the use of loop statements to solve iteration problems
CO3	Implement the concept of modular programming and recursion using functions
CO4	Implementation of decision statements and loops
CO5	Create a file and add, append or retrieve data using file handling

	CO/PO Mapping : (Strong(3)/Medium(2)/Weak(1) indicates strength of correlation):													
COs	Program Outcomes (PO's)/ Program Specific Outcomes (PSO's)													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	3	3	2	3	3	0	0	3	3	3	2	3	2
CO2	3	3	3	2	2	2	0	0	3	3	3	2	3	2
CO3	3	3	3	2	3	2	0	0	3	2	3	3	3	3
CO4	3	3	3	3	3	3	0	0	3	2	3	3	3	3
CO5	3	3	3	3	3	3	0	0	3	3	3	3	3	3

LIST OF PRACTICALS

1. Write down programs to implement various operators available in C.
2. Write down programs to implement the concept of conditional statements.
3. Write down programs to implement various loop statements.
4. WAP to implement the concept of break and continue statements.
5. Write down programs to implement functions using various parameter passing techniques.
6. WAP to implement the concept of recursion.
7. WAP to implement the concept of local and global variables.
8. Write down programs to implement array and its various operations.
9. Write down programs to implement strings and its various functions.
10. Write down programs to implement the concept of structure and union.
11. Write down programs to implement the concept of pointers.
12. Write down programs to implement the concept of file handling.

Note: The above-mentioned list of experiments is suggested list. Teacher may add more programs/ experiments as per requirement.

SEMESTER-II

Study Scheme and Syllabus for Five-Year Integrated M. Tech. Programme in Computer Science and Engineering

Title of the course : **Elements of Computer Engineering**

Subject code : **ESCS-401**

Weekly load : **2 Hours** **LTP 2-0-0**

Credits : **2**

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

CO1	Get acquainted with basics of computer system along with its various components
CO2	Know about various operating systems and memory
CO3	Study the C programming basics and learn the concept of operators
CO4	Understand the concept of decision statements and loops
CO5	Learn the use of functions, pointers, arrays, structures, union etc. for modular programming

	CO/PO Mapping : (Strong(3)/Medium(2)/Weak(1) indicates strength of correlation):													
COs	Program Outcomes (PO's)/ Program Specific Outcomes (PSO's)													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	1	3	1	3	0	0	0	3	1	3	2	3	3
CO2	3	1	3	3	3	0	0	0	3	2	3	2	3	3
CO3	3	2	3	1	3	3	0	0	2	2	3	2	3	3
CO4	3	2	3	3	3	3	0	0	3	3	3	3	3	3
CO5	3	3	3	3	3	3	0	0	3	3	3	3	3	3

Theory

Unit	Main Topics	Course outlines	Lecture (s)
Unit-1	1. Fundamentals of Computer Systems	Introduction and characteristics of computer systems, Generations, Classifications, Applications, Central Processing Unit, Memory, I/O devices, Introduction to Operating System and its types, Algorithm, Flowchart.	04
	2. C Programming Basics	Basic program construction, Structure of a C program, Compilation process, Preprocessor directives, Comments, Data types, Type conversions, Operators - Arithmetic, Relational, Logical, Conditional, Increment/decrement, Library functions, Header files.	04
	3. Loops and Decision Statements	*for* loop, *while* loop, *do* loop, Various forms of *if* statement,*switch* statement, *break* statement, *continue* statement, *goto* statement.	04
	4. Functions	Defining functions, Passing arguments to functions, Returning values from functions, Reference arguments, Storage classes.	04
Unit-2	5. Pointers	Pointers, Pointers to pointers, Declaring and initializing pointers, Pointer expressions, Pointers and arrays, Pointers and strings.	04
	6. Arrays	Arrays and strings, Declaring an array, Initializing arrays, Accessing the array elements, Working with multidimensional arrays, Declaring and initializing string variables, String handling functions.	04
	7. Structures	Declaring and initializing a structure, Accessing the	04

	and Unions	members of a structure, Nested structures, Array of structures, Using structures in functions, Pointers and structures, Declaring and initializing a union.	
	8. Files	Reading and writing to text and binary files, Character I/O, String I/O, File pointers, Error handling, Redirection, Command line arguments.	04

Total=32

Recommended Books:

1. Raja Raman V., Fundamentals of Computers, PHI.
2. Kernighan Brian W. and Ritchie, Dennis M., The C Programming language, Dorling Kingsley.
3. Balagurusamy E., Programming in ANSI C, TMH Publications

Study Scheme and Syllabus for Five-Year Integrated M. Tech. Programme in Computer Science and Engineering

Title of the course : **Elements of Computer Engineering Lab**

Subject Code : **ESCS-402**

Weekly load : **4 Hours** **LTP 0-0-4**

Credits : **2**

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

CO1	Solve basic mathematical problems using programming
CO2	Demonstrate the use of loop statements to solve iteration problems
CO3	Implement the concept of modular programming and recursion using functions
CO4	Implementation of decision statements and loops
CO5	Create a file and add, append or retrieve data using file handling

	CO/PO Mapping: (Strong(3)/Medium(2)/Weak(1) indicates strength of correlation):													
COs	Program Outcomes (PO's)/ Program Specific Outcomes (PSO's)													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	3	3	2	3	3	0	0	3	3	3	2	3	2
CO2	3	3	3	2	2	2	0	0	3	3	3	2	3	2
CO3	3	3	3	2	3	2	0	0	3	2	3	3	3	3
CO4	3	3	3	3	3	3	0	0	3	2	3	3	3	3
CO5	3	3	3	3	3	3	0	0	3	3	3	3	3	3

LIST OF PRACTICALS

1. Write down programs to implement various operators available in C.
2. Write down programs to implement the concept of conditional statements.
3. Write down programs to implement various loop statements.
4. WAP to implement the concept of break and continue statements.
5. Write down programs to implement functions using various parameter passing techniques.
6. WAP to implement the concept of recursion.
7. WAP to implement the concept of local and global variables.
8. Write down programs to implement array and its various operations.
9. Write down programs to implement strings and its various functions.
10. Write down programs to implement the concept of structure and union.
11. Write down programs to implement the concept of pointers.
12. Write down programs to implement the concept of file handling.

Note: The above-mentioned list of experiments is suggested list. Teacher may add more programs/ experiments as per requirement.

SEMESTER-III

Study Scheme and Syllabus for Five-Year Integrated M. Tech. Programme in Computer Science and Engineering

Title of the course : **Digital Electronics**

Subject Code : **PCCS-551**

Weekly load : **4 Hours** **LTP 3-1-0**

Credits : **4**

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

CO1	Learn various number systems, binary codes, and arithmetic operations used in digital electronics.
CO2	Simplify Boolean expressions and realize digital circuits using Boolean algebra and Karnaugh maps.
CO3	Understand the working of combinational logic circuits using logic gates.
CO4	Understand the working and characteristics of flip-flops and other sequential logic circuits.
CO5	Understand the basic operation of registers and counters using sequential circuit design techniques and state diagrams.

	CO/PO Mapping: (Strong(3)/Medium(2)/Weak(1) indicates strength of correlation):													
COs	Program Outcomes (PO's)/ Program Specific Outcomes (PSO's)													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	1	2	2	2	1	0	0	2	0	0	2	2	2
CO2	3	2	3	2	2	2	0	0	2	0	0	2	2	2
CO3	3	3	3	2	2	2	0	0	3	3	3	2	2	2
CO4	3	3	3	2	2	3	0	0	3	3	3	3	3	3
CO5	3	3	3	2	2	3	0	0	3	3	3	3	3	3

Theory

Unit	Main Topics	Course outlines	Lecture(s)
Unit-1	1. Introduction, Number Systems and Codes	Introduction to the concept of Digital Electronics, Number systems, Binary number system, Octal number system, Hexadecimal number system, Signed and unsigned numbers, Arithmetic using different number systems, Representation of binary numbers in signed magnitude, Signed 1's & 2's complement notation, Rules for addition and subtraction with complement.	07
	2. Introduction to Various Types of Codes	BCD, Parity, Gray Code, Excess-3, 8421 Code.	05
	3. Boolean Functions and Their Minimization	Simplification of Boolean functions using Boolean theorems, Canonical and standard forms (SOP and POS) for Boolean functions, Objectives of the minimization procedures, Karnaugh map method, Don't care conditions, Realization of Boolean functions using only NAND and NOR gates.	07
	4. Combinational Logic Circuits	Half and full adder; Half and full subtractor, Parity generator and checker, Code converters, Carry look ahead generator, Binary multiplier, Majority circuits, Magnitude comparator.	05

	using Discrete Logic Gates		
Unit-2	5. Combinational Logic using MSI Circuits	Binary parallel adder, BCD adder, Encoder, Priority encoder, decoder, Multiplexer and de-multiplexer circuits, Implementation of Boolean functions using decoder and multiplexer, ALU, BCD to 7-segment decoder.	07
	6. Introduction to Flip-flop Circuits	Basic concepts of sequential circuits, Cross-coupled SR flip-flop using NAND or NOR gates, D-type and Toggle flip-flops, JK flip-flop and race condition, Clocked flip-flops, Truth tables and excitation tables for flip-flops, Edge and level triggering, Master slave configuration, Edge triggered D flip-flop, Flip-flops with preset and clear.	07
	7. Sequential Logic Circuit Design and Counters	Sequential circuit, State table and state diagram, Design procedure, Basic concepts of counters and registers, Shift left and right register, Registers with parallel load, SIPO and PISO.	06
	8. Ripple (Asynchronous) Counters	Up-Down and Mod-N ripple counter, Design of synchronous counter using state diagrams and table, BCD counters, Modulo-N counter, Up-Down counter, Ring counter.	04

Total=48

Recommended Books:

1. Mano, M., Digital Logic and Computer Design, PHI.
2. Thomas, B., Digital Computer Fundamentals, McGraw-Hill.
3. Sandige, R., Modern Digital Design, McGraw-Hill.
4. Taub, Schilling, Digital Integrated Electronics, McGraw-Hill.
5. Fletcher, W. I., Engineering Approach to Digital Design, PHI.
6. Malvino, Leech, Digital Principles and Applications, TMH.
7. Wakerly, J. F., Digital Design: Principles and Practice Package, Pearson.

Study Scheme and Syllabus for Five-Year Integrated M. Tech. Programme in Computer Science and Engineering

Title of the course : Object Oriented Programming

Subject Code : PCCS-552

Weekly load : 4 Hours **LTP 3-1-0**

Credits : 4

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

CO1	Understand object-oriented programming fundamentals, contrast them with procedural paradigm, and implement classes using encapsulation, data hiding, and various class structures.
CO2	Manage object lifecycle using constructors and destructors, and build class hierarchies through inheritance, composition, and delegation.
CO3	Implement runtime polymorphism through virtual functions and abstract classes and extend language semantics via operator overloading and user-defined type conversions.
CO4	Apply dynamic memory management and exception handling techniques to prevent resource leaks, invalid references, and runtime failures.
CO5	Manage memory and pointers safely, handle exceptions, perform file I/O operations, and apply generic programming using templates and the Standard Template Library.

	CO/PO Mapping: (Strong(3)/Medium(2)/Weak(1) indicates strength of correlation):													
COs	Program Outcomes (PO's)/ Program Specific Outcomes (PSO's)													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	1	2	2	2	2	1	2	1	1	2	3	3	3	3
CO2	1	3	3	3	3	1	2	1	1	2	3	3	3	3
CO3	1	3	3	3	3	1	2	1	1	2	3	3	3	3
CO4	1	3	3	3	3	1	2	1	1	2	3	3	3	3
CO5	1	3	3	3	3	1	2	1	1	2	3	3	3	3

Theory

Unit	Main Topics	Course outlines	Lecture(s)
Unit-1	1. Introduction	Basic concepts of object-oriented programming, Characteristics of object-oriented programming, Comparison between procedural programming paradigm and object-oriented programming paradigm.	02
	2. Classes and Objects	Specifying a class, Creating class objects, Accessing class members, Access specifiers – public, private, and protected, Static members variables and functions, Static class objects, The const keyword, Friend functions, Friend classes, Empty classes, Nested classes, Local classes, Container classes.	08
	3. Constructors and Destruc	Need for constructors and destructors, Default constructor, Parameterized constructor, Copy constructor, Dynamic constructors, Destructors, Constructors and destructors with static members.	06

	tors		
	4. Inheritance	Introduction, Defining derived classes, Forms of inheritance, Ambiguity in multiple and multipath inheritance, Virtual base class, Object slicing, Overriding member functions, Object composition and delegation, Order of execution of constructors and destructors.	06
	5. Virtual Functions and Polymorphism	Concept of Binding – Early binding and late binding, Virtual functions, Pure virtual functions, Abstract classes, Virtual destructors & polymorphism.	06
	6. Operator Overloading and Type Conversion	Defining operator overloading, Rules for overloading operators, Overloading of unary operators and binary operators, Overloading of new and delete operators, Type conversion – Basic type to class type, Class type to basic type, Class type to another class type.	06
Unit-2	7. Pointers and Dynamic Memory Management	Understanding pointers, Accessing address of a variable, Declaring and initializing pointers, Accessing a variable through its pointer, Pointer arithmetic, Pointer to a pointer, Pointer to a function, Dynamic memory management – new and delete operators, Pointers and classes, Pointer to an object, Pointer to a member, this Pointer, Self-referential classes, Possible problems with the use of pointers - Dangling/wild pointers, Null pointer assignment, Memory leak and allocation failures.	06
	8. Exception Handling, Managing Data Files and Templates	Exception and its types, Exception handling mechanisms, File streams, Hierarchy of file stream classes, Error handling during file operations, Reading/Writing of files, Accessing records randomly, Updating files, Data formatting in memory buffers, Templates – Function Templates, Class Templates, Functors (Function objects), C++ Standard Library – Generic Programming, Standard Template Library.	06

Total=48

Recommended Books:

1. Lippman, S. B., Lajoie, J., and Moo, B. E., C++ Primer, Addison Wesley.
2. Stroustrup, Bjarne, The C++ Programming Language, Pearson Education.
3. Eckel, Bruce, Thinking in C++, Pearson.
4. Meyers, Scott, Effective Modern C++, O'Reilly.

Study Scheme and Syllabus for Five-Year Integrated M. Tech. Programme in Computer Science and Engineering

Title of the course : **Discrete Mathematics**

Subject Code : **PCCS- 553**

Weekly load : **4 Hours** **LTP 3-1-0**

Credits : **4**

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

CO1	Apply logical reasoning, set theory, and proof techniques to solve computational problems.
CO2	Analyze combinatorial and probabilistic problems using counting and probability principles.
CO3	Apply relations, functions, lattices, and Boolean algebra in computer science applications.
CO4	Use graph theory and recurrence relations to model and solve engineering problems.
CO5	Understand algebraic structures such as groups, rings, and fields for cryptographic and computational applications.

	CO/PO Mapping: (Strong(3)/Medium(2)/Weak(1) indicates strength of correlation):													
COs	Program Outcomes (PO's)/ Program Specific Outcomes (PSO's)													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	3	1	2	1	0	0	0	0	1	0	2	3	3
CO2	3	3	2	2	1	0	0	0	0	1	0	2	3	3
CO3	3	2	2	1	2	0	0	0	0	1	0	2	3	3
CO4	3	3	3	2	2	0	0	0	1	1	0	2	3	3
CO5	3	2	2	2	1	0	0	1	0	1	0	2	3	3

Theory:

Unit	Main Topics	Course outlines	Lecture(s)
Unit-1	1. Sets and Propositions	Introduction to Sets and Set Operations, Finite and Infinite Sets, Power Sets and Multisets, Ordered Pairs and Cartesian Product, Mathematical Induction, Principle of Inclusion and Exclusion, Propositions and Truth Tables, Logical Equivalence and Tautologies	06
	2. Basic Principles of Counting	Fundamental Principle of Counting, Addition and Multiplication Principles, Pigeonhole Principle and Applications, Permutations and Combinations, Binomial Coefficients, Generation of Permutations and Combinations, Introduction to Counting Trees and Decision Trees	07
	3. Basic Principles of Probability	Introduction to Discrete Probability, Conditional Probability, Bayes' Theorem, Random Variables, Information and Mutual Information	05
	4. Relations, Functions and Lattices	Binary Relations and their Properties, Equivalence Relations and Partitions, Partial Ordering Relations, Hasse Diagrams, Functions and Types of Functions, Recursive Functions, Introduction to Lattices, Sublattices, Distributive and Boolean Lattices, Job Scheduling Problem	07

Unit-2	5. Mathematical Logic	Logical Operators and Expressions, Propositional Algebra, Predicate Logic and Quantifiers, Rules of Inference, Proof Techniques: Direct Proof, Contradiction, Contrapositive	05
	6. Graphs and Planar Graphs	Introduction to Graphs, Basic Terminology, Directed and Undirected Graphs, Multigraphs and Weighted Graphs, Paths, Circuits and Connectivity, Shortest Paths in Weighted Graphs, Eulerian and Hamiltonian Graphs, Trees and Spanning Trees, Planar Graphs and Graph Coloring	06
	7. Recurrence Relations and Recursive Algorithms	Introduction to Recurrence Relations, Linear Recurrence Relations, Homogeneous and Non-Homogeneous Solutions, Particular and Total Solutions, Generating Functions, Recursive Definitions, Recursive Algorithms, Time Complexity using Recurrence Relations	06
	8. Groups, Rings and Boolean Algebra	Binary Operations, Semigroups and Monoids, Groups and Subgroups, Cyclic Groups, Rings and Fields, Modular Arithmetic, Principle of Duality, Boolean Algebra and Boolean Lattices, Boolean Functions and Logic Gates	06

Total=48

Recommended Books:

1. Liu, C.L., Elements of Discrete Mathematics, MGH.
2. Joshi, Maths Foundation of Discrete Mathematics, Wiley Eastern
3. Kolman, B., Busby, R. C., and Ross, S., Discrete Mathematical Structure, 5/E, PHI.
4. Deo, Narsingh, Graph theory with applications to Engineering & Computer Science, PHI.

Study Scheme and Syllabus for Five-Year Integrated M. Tech. Programme in Computer Science and Engineering

Title of the course : **Digital Electronics Lab**

Subject Code : **PCCS-554**

Weekly load : **2 Hours** **LTP 0-0-2**

Credit : **1**

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

CO1	Understand the operation and functionality of basic digital logic gates and verify their truth tables.
CO2	Analyze and simplify Boolean expressions using Boolean algebra and implement them using basic and universal gates.
CO3	Design and implement combinational logic circuits for arithmetic operations.
CO4	Develop and analyze data routing and code conversion circuits using combinational logic design techniques.
CO5	Analyze and implement basic sequential logic circuits using flip-flops.

	CO/PO Mapping: (Strong(3)/Medium(2)/Weak(1) indicates strength of correlation):													
COs	Program Outcomes (PO's)/ Program Specific Outcomes (PSO's)													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	2	3	2	2	2	0	0	2	2	2	2	2	2
CO2	3	2	2	2	2	2	0	0	2	2	2	2	2	2
CO3	3	2	2	1	2	2	0	0	3	2	2	3	3	2
CO4	3	3	3	2	2	2	0	0	3	3	3	3	3	3
CO5	3	3	3	2	2	2	0	0	3	3	3	3	3	3

LIST OF PRACTICALS

1. Study and verification of logic gates: AND, OR, NOT, NAND, NOR, and XOR for different input combinations using breadboard trainer kits.
2. Simplification of Boolean expressions and realization using basic/universal gates on breadboard trainer kits.
3. Design and realization of Half Adder and Half Subtractor using basic/universal gates on breadboard trainer kits.
4. Design and realization of Full Adder and Full Subtractor using basic/universal gates on breadboard trainer kits.
5. Design and realization of Encoder and Decoder circuits using basic/universal gates on breadboard trainer kits.
6. Design and realization of the following circuits using basic/universal gates on breadboard trainer kits: 4:1 Multiplexer (MUX) and 1:4 Demultiplexer (DEMUX).
7. Realization and verification of R-S Flip-Flop using basic/universal gates on breadboard trainer kits.
8. Realization and verification of J-K Flip-Flop using basic/universal gates on breadboard trainer kits.

Note: The above-mentioned list of experiments is suggested list. Teacher may add more programs/ experiments as per requirement.

Study Scheme and Syllabus for Five-Year Integrated M. Tech. Programme in Computer Science and Engineering

Title of the course : Object Oriented Programming Lab

Subject Code : PCCS-555

Weekly load : 2 Hours **LTP 0-0-2**

Credit : 1

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

CO1	Design classes with constructors, destructors, and access specifiers.
CO2	Build inheritance hierarchies and resolve ambiguity.
CO3	Implement polymorphism and overload operators.
CO4	Manage dynamic memory and handle exceptions.
CO5	Perform file I/O and use templates with STL.

CO/PO Mapping: (Strong(3)/Medium(2)/Weak(1) indicates strength of correlation):

COs	Program Outcomes (PO's)/ Program Specific Outcomes (PSO's)													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	1	2	2	2	2	1	2	1	1	2	3	3	3	3
CO2	1	3	3	3	3	1	2	1	1	2	3	3	3	3
CO3	1	3	3	3	3	1	2	1	1	2	3	3	3	3
CO4	1	3	3	3	3	1	2	1	1	2	3	3	3	3
CO5	1	3	3	3	3	1	2	1	1	2	3	3	3	3

LIST OF PRACTICALS

1. Implement classes with public, private, and protected members; demonstrate data hiding and access specifiers.
2. Use static data members and static member functions to track object count across multiple instances.
3. Write programs using friend functions and friend classes to access private members.
4. Implement nested classes, local classes, and container classes to show object composition.
5. Demonstrate default, parameterized, copy, and dynamic constructors.
6. Show the order of constructor and destructor execution in single, multilevel, and multiple inheritance.
7. Resolve ambiguity in multiple/multipath inheritance using scope resolution operator/virtual base classes.
8. Demonstrate object slicing and method overriding in inheritance hierarchies.
9. Implement runtime polymorphism using virtual functions, pure virtual functions, and abstract classes.
10. Show the effect of a virtual destructor when deleting a derived object through a base pointer.
11. Overload unary and binary operators for user-defined types.
12. Overload the new and delete operators to track memory allocation/deallocation.
13. Perform type conversions: basic type to class type, class type to basic type, and class type to another class type.
14. Use pointers (including pointer arithmetic, pointer to pointer, and pointer to function) and the this pointer.

15. Dynamically allocate and deallocate memory using new and delete; detect memory leaks, dangling pointers, and wild pointers.
16. Implement a self-referential class (e.g., a linked list node) and demonstrate basic list operations.
17. Handle runtime errors using try, throw, and catch with multiple catch blocks and user-defined exceptions.
18. Perform file operations: write, read, update, and randomly access records using file streams.
19. Use string streams for data formatting in memory buffers.
20. Write function templates and class templates for generic data structures (e.g., a stack or queue).
21. Implement functors (function objects) and use them with STL algorithms.
22. Write small programs using STL containers (vector, list, map) and iterators.

Note: The above-mentioned list of experiments is suggested list. The teacher may add more programs/ experiments as per requirement.

SEMESTER-IV

Study Scheme and Syllabus for Five-Year Integrated M. Tech. Programme in Computer Science and Engineering

Title of the course : **Computer Organization and Architecture**

Subject Code : **PCCS-561**

Weekly load : **4 Hours** **LTP 3-1-0**

Credits : **4**

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

CO1	Understand the fundamentals of computer organization, architecture, data representation, and register transfer language concepts.
CO2	Analyze micro-operations, instruction formats, instruction cycles, interrupts, and the design of a basic computer system.
CO3	Apply concepts of CPU organization, addressing modes, stack organization, pipelining, and parallel processing in processor design.
CO4	Compare control organization techniques including hardwired and micro-programmed control units and microprogramming methods.
CO5	Understand memory hierarchy, cache organization, virtual memory, paging, segmentation, and input/output organization.

	CO/PO Mapping: (Strong(3)/Medium(2)/Weak(1) indicates strength of correlation):													
COs	Program Outcomes (PO's)/ Program Specific Outcomes (PSO's)													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	2	2	2	2	2	2	0	0	2	2	2	2	2	2
CO2	2	2	2	2	2	2	0	0	2	2	2	2	3	2
CO3	1	2	2	2	2	2	0	0	2	2	2	2	2	2
CO4	3	2	2	2	3	2	0	0	2	3	2	2	2	3
CO5	2	2	2	2	2	2	0	0	2	3	2	2	2	2

Theory

Unit	Main Topics	Course Outlines	Lecture(s)
Unit-1	1. Introduction	Computer architecture, organization and designing, Historical architectural trends, Data representation: Number system and conversion, Compliments, Fixed point representation, Floating point representation, Computer arithmetic (addition, subtraction and multiplication) Introduction to GPUs, CPU vs GPU architecture, GPU components, Applications of GPUs	06
	2. Register Transfer Language and Micro-Operations	Register transfer language, Bus and memory transfer, Arithmetic, logic and shift micro-operations, Constructing an arithmetic logic shift unit	06
	3. Basic Computer Architecture and Design	Computer registers, Instructions, Instruction codes, Instruction formats, Instruction classifications, Memory reference instructions, Register reference instructions, Input/output and interrupts, Instruction set completeness, Instruction execution cycle, Interrupt cycle, Timing & control unit, Complete computer description and design	07

		of basic computer	
	4. Central Processing Unit	General register organization, Stack organized CPU, Addressing modes, Data transfer and manipulation, RISC Vs CISC, Introduction to parallel processing and pipeline processing	05
Unit-2	5. Control Organization	Hardwired and micro-programmed control organization, Horizontal and vertical microprogramming	06
	6. Input Output Organization	I/O interfacing, Asynchronous data transfer, Modes of transfer, Priority interrupt, DMA, Input output processor	06
	7. Memory Organization	Memory Systems: Principle of locality, Principles of memory hierarchy, Caches, Associative memory, Main memory, Virtual memory, Paging and segmentation	06
	8. Case Studies	Case studies of contemporary architecture for processors of families like Intel and AMD, NVIDIA GPU architecture	06

Total=48

Recommended Books:

1. Hamacher, Vranesic, Computer Organization, McGraw Hill.
2. Stallings, W., Computer Organization, TMH.
3. Hayes, John P., Computer Architecture and Organization, McGraw Hill.
4. Carpirdli, Computer Architecture & Organization, Pearson Education Asia.
5. Mano, M., Computer Architecture & Organization, PHI.

Study Scheme and Syllabus for Five-Year Integrated M. Tech. Programme in Computer Science and Engineering

Title of the course : **Operating Systems**

Subject Code : **PCCS-562**

Weekly load : **3 Hours** **LTP 3-0-0**

Credits : **3**

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

CO1	Learn basic concepts of operating systems along with implementation of scheduling algorithms in process management.
CO2	Analyze critical section problem in inter-process communication and use of memory management techniques.
CO3	Implement page replacement algorithms and use virtual memory concepts.
CO4	Know about file structure, file management and disk management.
CO5	Learn the concept of deadlock and implement various algorithms used for its detection and recovery.

	CO/PO Mapping: (Strong(S)/Medium(M)/Weak(W) indicates strength of correlation):													
COs	Program Outcomes (PO's)/ Program Specific Outcomes (PSO's)													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	2	3	1	3	2	0	0	2	2	2	2	2	2
CO2	3	2	2	2	2	2	0	0	2	3	3	2	3	3
CO3	3	3	2	2	0	3	0	0	3	2	2	2	3	2
CO4	3	2	2	2	2	2	0	0	3	3	2	2	2	2
CO5	3	2	1	2	0	2	0	0	3	2	2	2	3	3

Theory

Unit	Main Topics	Course outlines	Lecture(s)
Unit-1	1. Basic Concept of Operating Systems	Evolution of operating systems, Operating system classifications, Structure of an operating system, Types of operating systems (batch, multiprogramming, multiprocessing, time-sharing, real time systems etc.), Concept of virtual machine, Case study on UNIX and Windows operating systems.	06
	2. Process Management	Process overview, Process states, PCB, Levels of scheduler and CPU scheduling algorithms, Multiple-processor scheduling, Scheduling criteria, Threads, Process scheduling objects and techniques.	06
	3. Inter-process Communication	Concurrent processes, The critical section and mutual exclusion problem: algorithms, Semaphores, Critical region, Conditional critical region, Race condition, Monitors, Message passing: examples in contemporary OS, Classical process co-ordination problems.	06
	4. Memory Management	Memory hierarchy, Static and dynamic memory allocation, Overview of swapping, Multiple partitions, Contiguous and non-contiguous memory allocation, Internal and external fragmentation, Concepts of paging, Segmentation.	06
Unit-2	5. Virtual Memory	Virtual Memory Concepts, Demand paging, Performance fragmentation and compaction, Concept	08

		of page fault and dirty page/dirty bit, Page replacement and allocation algorithms, Memory protection, System calls, Linux/Windows virtual memory techniques.	
	6. File Management	File concepts, Access methods, Directory structure, File protection, File System structure, Allocation methods, Secondary storage management: disk structure, Disk scheduling algorithms, Disk management, Swap-space management, Disk reliability.	08
	7. Deadlocks	Introduction to deadlocks, Necessary and sufficient conditions, Deadlock prevention and avoidance, Banker's algorithm, Deadlock detection and recovery.	08

Total=48

Recommended Books:

1. Silberschatz, A., and Galvin, Operating System Concepts, John Wiley & Sons.
2. Stallings, W., Operating Systems: Internals and Design Principles, Pearson.
3. Tanenbaum, Andrew S., Operating Systems - Design and Implementation, PHI.
4. Crawley, Operating Systems - An Object Oriented Approach, McGraw Hill.

Study Scheme and Syllabus for Five-Year Integrated M. Tech. Programme in Computer Science and Engineering

Title of the course : Data Structures and Algorithms

Subject Code : PCCS-563

Weekly load : 4 Hours LTP 3-1-0

Credits : 4

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

CO1	Understand various techniques for the representation of data in the real world.
CO2	Design and implement various data structures and algorithms.
CO3	Be familiar with the utilization of data structures in problem-solving.
CO4	Learn how to analyse the time and space requirements of a given algorithm.
CO5	Develop applications using various data structure algorithms.

	CO/PO Mapping: (Strong(3)/Medium(2)/Weak(1) indicates strength of correlation):													
COs	Program Outcomes (PO's)/ Program Specific Outcomes (PSO's)													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	3	3	3	2	0	0	0	2	0	0	0	1	0
CO2	3	3	3	3	3	2	0	0	3	3	3	2	3	2
CO3	3	3	3	3	3	2	0	0	3	3	3	3	3	3
CO4	3	3	3	3	3	2	0	0	3	3	2	2	3	3
CO5	3	3	3	3	3	3	0	0	3	3	2	3	3	3

Theory

Unit	Main Topics	Course Outlines	Lecture(s)
Unit-1	1. Introduction	Basic concepts and notations, Data structures and data structure operations, Mathematical notation and functions; Algorithmic complexity, Asymptotic Notations (Big-O, Big-Ω, Big-Θ), Master's Theorem, Time-space trade-off.	04
	2. Arrays	Introduction to an array, Representation of an array, One-dimensional arrays, Multidimensional arrays, Address calculation, String processing, Memory representation of a string, Operations on a string, Pattern matching.	08
	3. Linked Lists	Introduction, Basic concepts of linked list – memory representation, building a linked list, traversing, insertion, deletion, searching, Doubly linked list, Merging two lists, Header linked list, Circular linked list.	06
	4. Stacks and Queues	Introduction to stacks, Representation of stack, Implementation of stack, Polish notation, Introduction to queues, Implementation of queues, Circular queues, Double-ended queues, Priority queues.	06
Unit-2	5. Trees	Binary trees, Complete binary trees, Extended binary trees, Binary search trees – representation, traversal, insertion, deletion, Heaps, Balanced binary trees – AVL trees, RB Trees, B-trees, Applications of trees.	06
	6. Graphs	Basic concepts and definitions, Representation of graphs – adjacency list, adjacency matrix, path matrix, Graph traversal – BFS, DFS, Shortest path algorithms – Dijkstra's algorithm, Topological Sorting, Prim's algorithm, Kruskal's algorithm.	06

	7. Sorting and Searching	Linear search, Binary search, Bubble sort, Insertion sort, Quick sort, Selection sort, Merge sort, Heap sort, Hashing techniques.	06
	8. Introduction to Algorithm Design Techniques	Divide and conquer – Towers of Hanoi problem, Greedy method, Dynamic programming, Backtracking, Recursion	06

Total=48

Recommended Books:

1. Thareja, R., Data Structures Using C, Oxford University Press, 2nd Edition, 2014.
2. Horowitz, E., Sahni, S., Anderson-Freed, S., Fundamentals of Data Structures in C, Universities Press, 2nd Edition, 2008.
3. Weiss, M.A., Data Structures and Algorithm Analysis in C, Pearson Education, 2nd Edition, 1997.
4. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C., Introduction to Algorithms, MIT Press, 4th Edition, 2022.
5. Srivastava, S.K., Srivastava, D., Data Structures Through C in Depth, BPB Publications, Revised Edition, 2011.

Study Scheme and Syllabus for Five-Year Integrated M. Tech. Programme in Computer Science and Engineering

Title of the course : **Operating Systems Lab**

Subject Code : **PCCS-564**

Weekly load : **2 Hours** **LTP 0-0-2**

Credit : **1**

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

CO1	Apply Linux/Unix commands and shell scripting techniques to perform file handling and process-related operations in an operating system environment.
CO2	Demonstrate process management concepts by implementing CPU scheduling, process creation, and inter-process communication techniques.
CO3	Analyze synchronization and concurrency problems using threads, semaphores, and deadlock handling mechanisms.
CO4	Implement memory and storage management techniques including paging, page replacement, disk scheduling, and file allocation methods.
CO5	Develop and evaluate operating system-based programs using system calls and simulation techniques to solve real-world computing problems.

CO/PO Mapping: (Strong(S)/Medium(M)/Weak(W) indicates strength of correlation):														
COs	Program Outcomes (PO's)/ Program Specific Outcomes (PSO's)													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	2	2	2	1	0	0	0	1	0	0	2	2	2
CO2	3	3	3	2	3	2	0	0	3	3	3	2	3	3
CO3	3	3	3	2	3	2	0	0	2	2	3	2	3	3
CO4	3	3	3	3	3	3	0	0	3	2	2	3	3	2
CO5	3	3	3	3	3	3	0	0	3	3	3	3	3	3

LIST OF PRACTICALS

1. Study of Linux/Unix Operating System Commands: ls, pwd, mkdir, rm, cp, mv, cat, grep, chmod, ps, top.
2. Shell Programming Basics: Write shell scripts for arithmetic operations, file handling, and decision-making.
3. Implementation of CPU Scheduling Algorithms:
    a. First Come First Serve (FCFS)
    b. Shortest Job First (SJF)
    c. Priority Scheduling
    d. Round Robin Scheduling
4. Process Creation and Management: Implementation of programs using system calls such as fork(), exec(), wait().
5. Inter-Process Communication (IPC): Implementation of programs using:
    a. Pipes
    b. Message Queues
    c. Shared Memory
6. Thread Programming: Create and manage threads using POSIX Threads (pthread).
7. Implementation of the Producer–Consumer Problem: Implementation of synchronization using semaphores.

8. Implementation of the Reader–Writer Problem: Implementation using synchronization techniques.
9. Deadlock Avoidance Algorithm: Implementation of Banker's Algorithm.
10. Deadlock Detection: Simulating deadlock detection methods.
11. Memory Management Techniques: Simulation of:
    a. Paging
    b. Segmentation
12. Page Replacement Algorithms: Implementation of:
    a. FIFO
    b. LRU
    c. Optimal Page Replacement
13. Disk Scheduling Algorithms: Implementation of:
    a. FCFS
    b. SSTF
    c. SCAN
    d. C-SCAN
14. File Allocation Methods: Sequential, Indexed, and Linked allocation simulation.
15. File Management System: Create, delete, read, and update files.
16. System Calls in Linux: Demonstration of file and process system calls.
17. Disk Space Management: Study of fragmentation and storage allocation.
18. Mini Project: Simple command interpreter/shell or process scheduling simulator.

Note: The above-mentioned list of experiments is suggested list. The teacher may add more programs/ experiments as per requirement.

Study Scheme and Syllabus for Five-Year Integrated M. Tech. Programme in Computer Science and Engineering

Title of the course : **Data Structures and Algorithms Lab**

Subject Code : **PCCS-565**

Weekly load : **2 Hours** **LTP 0-0-2**

Credit : **1**

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

CO1	Understand the fundamental concepts of data structures, algorithms, and complexity analysis.
CO2	Design, implement, test, and debug programs using various data structures, including arrays, linked lists, stacks, queues, trees, and graphs.
CO3	Apply suitable data structures and algorithmic techniques for efficient problem-solving.
CO4	Implement and analyze standard searching, sorting, graph traversal, and tree traversal algorithms.
CO5	Develop applications using advanced data structures and algorithm design techniques for real-world computing problems.

CO/PO Mapping: (Strong(S)/Medium(M)/Weak(W) indicates strength of correlation):

COs	Program Outcomes (PO's)/ Program Specific Outcomes (PSO's)													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	2	2	2	1	0	0	0	1	0	0	2	2	2
CO2	3	3	3	2	3	2	0	0	3	3	3	2	3	3
CO3	3	3	3	2	3	2	0	0	2	2	3	2	3	3
CO4	3	3	3	3	3	3	0	0	3	2	2	3	3	2
CO5	3	3	3	3	3	3	0	0	3	3	3	3	3	3

LIST OF PRACTICALS

1. WAP to perform insertion, deletion, and traversal operations on one-dimensional arrays.
2. WAP to implement operations on multidimensional arrays.
3. WAP for string manipulation operations and pattern matching.
4. WAP to implement a singly linked list with insertion, deletion, and searching operations.
5. WAP to implement doubly linked list operations.
6. WAP to implement circular linked list operations.
7. WAP to merge two linked lists.
8. WAP to implement stack operations using arrays and linked lists.
9. WAP to convert an infix expression into a postfix expression using a stack.
10. WAP to implement queue and circular queue operations.
11. WAP to implement a double-ended queue (deque).
12. WAP to implement priority queue operations.
13. WAP to implement a binary search tree to perform tree traversals using inorder, preorder, and postorder techniques, insertion and deletion operations.
14. WAP to implement AVL tree operations.
15. WAP to implement graph traversal using Breadth First Search (BFS).
16. WAP to implement graph traversal using Depth First Search (DFS).
17. WAP to implement Dijkstra's shortest path algorithm.
18. WAP to implement Minimum Spanning Tree using Prim's or Kruskal's algorithm.
19. WAP to implement searching algorithms: linear search and binary search.

20. WAP to implement sorting algorithms: bubble sort, insertion sort, quick sort, merge sort, and heap sort.

Note: The above-mentioned list of experiments is a suggested list. The teacher may add more programs/ experiments as per the requirement.