SCHEME FOR EXAMINATION
B.TECH (FOUR YEAR) DEGREE COURSE
SECOND YEAR, INFORMATION TECHNOLOGY
SEMESTER III
EFFECTIVE FROM SESSION 2021-22

SL. NO.	SUBJECT CODE	SUBJECTS	PERIODS/ WEEK			EVALUATION SCHEME			CREDITS
			L	T	P	IA	ESE	TOTAL	
THEORY									
1	IT203TES06	ANALOG ELECTRONIC CIRCUITS	3	0	0	30	70	100	3
2	IT203TPC01	DATA STRUCTURE & ALGORITHMS	3	0	0	30	70	100	3
3	IT203TPC02	DIGITAL ELECTRONICS	3	0	0	30	70	100	3
4	IT203TBS05	MATHEMATICS-III	3	1	0	30	70	100	4
5	IT203TPC03	OBJECT ORIENTED PROGRAMMING	3	1	0	30	70	100	4
PRACTICAL									
1	IT203PES06	ANALOG ELECTRONIC CIRCUITS LAB	0	0	4	30	20	50	2
2	IT203PPC01	DATA STRUCTURE LAB	0	0	4	30	20	50	2
3	IT203PPC02	DIGITAL ELECTRONICS LAB	0	0	4	30	20	50	2
4	IT203PPC03	OBJECT ORIENTED PROGRAMMING LAB		0	4	30	20	50	2
TOTAL CREDITS									**25**
IA- INTERNAL ASSESSMENT, ESE-END SEMESTER EXAMINATION, L-LECTURE, T-TUTORIAL, P-PRACTICAL									

SCHOOL OF STUDIES OF ENGINEERING & TECHNOLOGY
GURU GHASIDAS VISHWAVIDYALAYA, BILASPUR (C.G.) (A CENTRAL UNIVERSITY)
B. TECH. INFORMATION TECHNOLOGY

Programme Outcomes: Graduates will be able to:

PO1: Fundamentals: Apply knowledge of mathematics, science and engineering.

PO2: Problem analysis: Identify, formulate and solve real time engineering problems using first principles.

PO3: Design: Design engineering systems complying with public health, safety, cultural, societal and environmental considerations

PO4: Investigation: Investigate complex problems by analysis and interpreting the data to synthesize valid solution.

PO5: Tools: Predict and model by using creative techniques, skills and IT tools necessary for modern engineering practice.

PO6: Society: Apply the knowledge to assess societal, health, safety, legal and cultural issues for practicing engineering profession.

PO7: Environment: Understand the importance of the environment for sustainable development.

PO8: Ethics: Apply ethical principles and commit to professional ethics, and responsibilities and norms of the engineering practice.

PO9: Teamwork: Function effectively as an individual and as a member or leader in diverse teams and multidisciplinary settings.

PO10: Communication: Communicate effectively by presentations and writing reports.

PO11: Management: Manage projects in multidisciplinary environments as member or a team leader.

PO12: Life-long learning: Engage in independent lifelong learning in the broadest context of technological change.

Programme Specific Outcomes:

PSO1: To apply knowledge of recent computing technologies, skills and current tools of Information Technology Engineering.

PSO2: To design software systems, components or processes to meet identified needs within economic, environmental and social constraints.

PSO3: To explore research gaps, analyze and carry out research in the specialized/emerging areas.

SUB CODE	L	T	P	DURATION	IA	ESE	CREDITS
IT203TES06	**3**	**0**	**0**	**3 HOURS**	**30**	**70**	**3**

ANALOG ELECTRONIC CIRCUITS

Course Objective

1. To understand the operation of the various circuits of low frequency model of transistors and FET circuits and its parameter.
2. To apply concepts for the design of high frequency Amplifiers and multi stage amplifiers. To apply concept of Millers Theorem.
1. To understand the effects of negative feedback on amplifier circuits.
2. To understand concept of the different oscillator circuits for waveform Generation and determine frequency of oscillation.
3. To understand concept of the operation of various types of power amplifier circuits and tuned amplifiers.

UNIT- I

Low frequency transistor, amplifier, graphical analysis of CE Amplifier, h-parameter models for CB, CE, CC configurations and their interrelationship, analysis and comparison of the three configurations, linear analysis of transistor circuits, Miller's Theorem: Cascading, simplified models and calculation of CE and CC Amplifiers, effect of emitter resistance in CE amplifiers, cascade amplifiers, Darlington pair, analysis of single stage FET amplifiers-CS and CD configuration.

UNIT II

High frequency transistor amplifier, CE hybrid pi model, validity and parameter variation, current gain with resistive load, frequency response of a single stage CE amplifier, gain bandwidth product, CC stage high frequencies, multistage amplifier, classification, distortion in amplifiers, frequency response, bode plots, step response, pass band of cascaded stages, response of a two stage RC coupled amplifier at low and high frequencies, sources of noise in transistor circuits, noise figure.

UNIT III

Feedback Amplifiers: Classification, feedback concept, ideal feedback amplifier, properties of negative feedback amplifier topologies: method of analysis of feedback amplifier, voltage series feedback, voltage series feedback pair, current series, current shunt, voltage shunt feedback, effect of feedback on amplifier bandwidth and stability.

UNIT IV

Large Signal/power amplifier, classification, large signal amplifier characteristics, class A amplifiers, class A amplifier with direct coupled resistive load, transformer coupled class A amplifier, class A push pull amplifiers, class B amplifiers, transformer coupled push pull class B amplifier, complementary symmetry push pull class B amplifier, class AB amplifier, class C amplifier, Harmonic Distortion, Push Pull Amplifiers, Cross over Distortion.

UNIT V

Oscillator: Sinusoidal oscillator, phase shift oscillator, Wien bridge oscillator, Resonant circuit oscillator, LC Collpit, LC Hartley, Amplitude, Frequency, and phase stability analysis of all oscillators, General form of oscillator configuration, crystal oscillator, tuned Amplifiers, classification of tuned amplifier, analysis of single and double tuned amplifiers, stagger tuned amplifier.

SUGGESTED TEXT BOOKS:

1. Integrated Electronics, Millman & Halkias, TMH
2. Microelectronics, Millman & Grabel, TMH

REFERENCE BOOKS:

1. Electronic Device & Circuits, David A Bell, PHI
2. Electronic Device & Circuits Theory, Boylestad & Nashelsky, PHI

Course Outcomes

1. Design and Implement various Low frequency transistors and FET models and Compare various transistors configuration
2. Design and analyze different configuration of High frequency BJT models
3. Design and analyze various types of feedback amplifier circuits
4. Design and Implement of various sinusoidal and non-sinusoidal oscillators.
5. Analyze different types of power amplifiers and Tuned amplifiers

CO	PO											PSO		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	3	3	3	3							3	3	3	3
CO2	3	3	3	3	3							3	3	3	3
CO3	3	3	3	2	2							3	3	3	2
CO4	3	2	3	2	2							3	3	3	2
CO5	3	2	3	2	1							3	3	3	2

Weightage: 1-Sightly, 2-Moderately, 3-Strongly

SUB CODE	L	T	P	DURATION/WEEK	IA	ESE	CREDITS
IT203TPC01	3	0	4	3 HOURS	30	70	3

DATA STURCTURE & ALGORITHMS

Course Objective

- CO1 - To impart the basic concepts of data structures and algorithms and understand concepts about searching and sorting techniques.
- CO2 - To understand basic concepts about Linked lists and master the implementation of linked data structures.
- CO3 - To understand basic concepts about stacks and queues.
- CO4 - To understand basic concepts about Tree.
- CO5 - To understand basic concepts about Graph and be familiar with some graph algorithms such as shortest path and minimum spanning tree.

Course Outcome

Upon completion of this course, the students will be able to

- Student will be able to choose appropriate data structure as applied to specified problem definition.
- Student will be able to handle operations like searching, insertion, deletion, traversing mechanism etc. on various data structures.
- Students will be able to apply concepts learned in various domains like DBMS, compiler construction etc.
- Students will be able to use linear and non-linear data structures like stacks, queues, linked list etc.
- Students will be able to know about different types of graphs and their applications.

UNIT- I

Introduction: Basic Terminology, Definition of Data Structure, Types of Data Structure, Operation on Data Structure, **Arrays:** Array Definition, Representation of Arrays: Row Major Order, and Column Major Order.
Searching and Sorting: Selection Sort, Insertion Sort, Bubble Sort, Quick Sort, Merge Sort, Binary Search, Linear Search.

UNIT II

Linked lists: Definition, Representation and Implementation of Singly Linked Lists, Traversing and Searching of Linked List, Insertion and deletion to/from Linked Lists, Insertion and deletion Algorithms, Doubly Linked List, Circularly Linked List.

UNIT III

Stacks: Array Representation and Implementation of stack, Operations on Stacks: Push & Pop, Array Representation of Stack, Linked Representation of Stack, Operations Associated with Stacks, Application of stack: Conversion of Infix to Prefix and Postfix Expressions, Evaluation of postfix expression using stack.,

Queue: Array and linked representation of queues, Operations on Queue: Create, Add, Delete, Full and Empty, Circular queues, Deques.

UNIT IV

Trees: Basic Technology , Binary Tree , Binary tree representation , Algebraic Expressions , Complete Binary Tree, Extended Binary Tree, Full Binary Tree, Array and linked Representation of Binary trees, Traversing Binary trees, Threaded Binary trees, Binary search trees (BST), Insertion and deletion in BST, AVL trees, Heap and heap sort.

UNIT V

Graph: Terminology & Representations, Graphs & Multi-graphs, Directed Graphs, Weighted Graph, Sequential Representations of Graphs, Adjacency Matrices, Adjacency List, Path Matrices, Linked Representations of Graphs, Graph Traversal - DFS, BFS, Shortest Path algorithm: Warshal Algorithm and Dijikstra Algorithm, Spanning Trees, Minimum Cost Spanning Trees: Prims and Kruskal algorithm.

References books:

1. Lipschutz, "Data Structures with C" Schaum's Outline Series, TMH.
2. Horowitz and Sahani, "Fundamentals of data Structures", Galgotia Publication Pvt. Ltd.
3. R. Kruse etal, "Data Structures and Program Design in C", Pearson Education Asia.
4. A. M. Tenenbaum, "Data Structures using C & C++", Prentice-Hall of India Pvt. Ltd.
5. K Loudon, "Mastering Algorithms with C", Shroff Publisher & Distributors Pvt. Ltd.
6. Jean Paul Trembley and Paul G. Sorenson, "An Introduction to Data Structures with applications", McGraw Hill.
7. G A V Pai, "Data Structures and Algorithms", TMH.
8. G.S.Baluja, "Data Structures through C", Dhanpat Rai & Co.
9. Yashavant Kanetkar, "Data Structure Through C", BPB Publication.

CO	PO												PSO		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	1	3	3	2	2								3	3	3
CO2	2	3	1	2	3								3	3	3
CO3	3	3	3	3	3								3	3	3
CO4	2	3	2	3	3								3	3	3
CO5	3	2	1	2	3								3	3	3

Weightage: 1-Sightly, 2-Moderately, 3-Strongly

SUB CODE	L	T	P	DURATION	IA	ESE	CREDITS
IT203TPC02	3	0	0	3 HOURS	30	70	3

DIGITAL ELECTRONICS

Course Objectives:

1. To understand the basic knowledge of digital logic and components.
2. To simplify the Boolean expressions or Combinational circuits for compact circuits.
3. Design of combinational circuits and sequential circuits.
4. Application of knowledge to understand digital electronics circuits.
5. To impart how to design Digital Circuits.

Course Outcome (COs):

At the end of this course, students will demonstrate the ability to

1. Employ the codes and number systems converting circuits and Compare different types of logic families which are the basic unit of different types of logic gates in the domain of economy, performance and efficiency.
2. Analyze different types of digital electronic circuit using various mapping and logical tools and know the techniques to prepare the most simplified circuit using various mapping and mathematical methods.
3. Design different types of with and without memory element digital electronic circuits for particular operation, within the realm of economic, performance, efficiency, user friendly and environmental constraints.
4. Apply the fundamental knowledge of analog and digital electronics to get different types analog to digitalized signal and vice-versa converters in real world with different changing circumstances.
5. Assess the nomenclature and technology in the area of memory devices and apply the memory devices in different types of digital circuits for real world application.

UNIT 1 - Fundamentals of Digital systems and logic families

Digital signals, digital circuits, AND,OR, NOT, NAND, NOR and Exclusive OR operations, Boolean algebra, examples of IC gates, number systems- binary, signed binary, octal, Hexadecimal number, binary arithmetic, Once's and two's complements, arithmetic codes, error detecting, and correcting codes, characteristics of digital ICs, digital logic families, TTL, schottky TTL and CMOS logic , interfacing CMOS and TTL, Tri-state logic.

UNIT 2 - Combinational Digital Circuits

Standard representation for logic function, K map representation, simplification of logic functions, using K map, minimization of logical functions. Don't care conditions, Multiplexes, De- Multiplexes, / Decoders, Adders, Sub tractors, BCD arithmetic, carry look ahead, serial adders , ALU , elementary ALU design, popular MSI chips , digital comparator , parity checker,

/ generator, code converters, priority encoders, decoders/ drivers, for display devices, Q-M method of function realization.

UNIT 3 - Sequential circuits and systems

A 1 bit memory, the circuits properties, of Bi-stable latch, the clocked SR flip flop, JK flip flops, T flip flops, D flip flops, applications of flip flops, shift registers, applications of shift registers, serial to parallel converter, parallel to serial converter, ring counter, sequence generator, ripple (Asynchronous) counters, synchronous counters, counter's design using flip flops, special counter IC's, Asynchronous sequential counters, applications of counters.

UNIT 4 - A/D and D/A converters

Digital to analog converters: weighted registers/ converters, R-2R Ladder, D/A converters , specifications for D /A converters, examples of D /A converter ICs, sample and hold circuits, Analog to digital converters: quantization and encoding, parallel comparator , A/D converter, successive approximation A/D converter, counting A/D converter, dual slop A / D converter, A/D converter using voltage to frequency and voltage to time conversion, specifications of A/D converters, example of A/D converter ICs.

UNIT 5 - Semiconductor memories and Programmable logic devices

Memory organization and operation, expanding memory size, classification and characteristics of memories, Sequential memories, read-only memory(ROM), read and write memory (RAM), content addressable memory (CAM) , charge coupled device memory (CCD) , commonly used memory chips, ROM as a PLD, Programmable logic array, Programmable array logic, complex Programmable logic devices (CPLDS) , Field Programmable Gate Array (GPGA).

Text / References:

1. M.M Mano, "Digital logic and Computer design", Pearson Education India.
2. R.P. Jain, " Modern Digital Electronics" , McGraw Hill Education.
3. A kumar, "Fundamentals of Digital Circuits ", Prentice Hall India.
4. S Salivahanan and S Arivazhagan" Digital Circuits and Design" OXFORD University Press.

CO	PO												PSO		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	3	2	-	3	-	-	-	-	-	-	3	3	2	1
CO2	3	3	3	3	-	-	-	-	-	-	-	3	3	3	1
CO3	3	3	3	2	-	2	-	2	-	-	1	3	3	3	1
CO4	3	3	3	3	3	2	-	2	2	3	3	3	3	3	1
CO5	3	3	2	2	3	-	-	2	2	3	3	3	3	3	1

Weightage: 1-Sightly, 2-Moderately, 3-Strongly

SUB CODE	L	T	P	DURATION/WEEK	IA	ESE	CREDITS
IT203TBS05	3	1	0	4 HOURS	30	70	4

MATHEMATICS III

Course Objective:

1. The primary objective of the course is to develop the basic understanding of numerical algorithms and skills to implement algorithms to solve mathematical problems on the computer.
2. Adequate exposure to learn alternative methods and analyze mathematical problems to determine the suitable numerical techniques.
3. Grasp the basic elements of numerical methods and error analysis.
4. The main objective of this course is to provide students with an introduction to the field of numerical analysis. Aside from developing competency in the topics and emphases listed above, the course aims to: further develop.
5. To apply problem solving skills through the introduction of numerical methods; provide a ground for applying knowledge acquired in previous mathematics courses; and give students an opportunity to develop and present an independent project.

UNIT-I Introduction of Errors and their Analysis, types of errors, numerical problems on error analysis, curve fitting: method of least squares, fitting of exponential curves $y=ae^{bx}$ fitting of the curve $y = ab^x$ fitting of the curve $y = ax^b$ at Method or moments

UNIT-II Numerical Solution of Algebraic and Transcendental Equations: Graphical method bisection Method, Secant Method, Regula-falsi Method, Newton Raphson Method, Solution of a system of a simultaneous linear algebraic Equations Direct methods: Gauss elimination Method, Gauss Jordan method, Iterative methods, Jacobi Iterative Method. Gauss Seidel Iterative method.

UNIT-III The Calculus of Finite Differences: Finite differences. Difference formula operators and Relation between operators. Inverse Operator, Interpolation with equal intervals: -Newton's forward and backward interpolation formula, Interpolation with Unequal intervals: - Lagrange's interpolation Newton's difference formula, inverse interpolation.

UNIT- IV Numerical Differentiation and Integration: - Numerical Differentiation Newton's forward and Backward difference interpolation formula, Maxima and Minima of a Tabulated function, Numerical Integration :- Trapezoidal rule, simpson's (1/3)rd and (3/8)th rule, Boole's Rule, weddle rule.

Difference Equations: Definition, order and degree of a difference equation. Linear difference Equations, Difference equations reducible to linear form, simultaneous difference equations with Constant coefficients.

UNIT- V Numerical solution of ordinary differential equation: Taylor series method, Euler's Method, Modified Euler method Runge's method, Runge-Kutta method, Numerical method for solution of partial differential equations, General linear partial differential equation, Laplace equation and Poisson equation.

Text Books:

1. JAIN & IYNGAR Numerical Methods for Scientific and Engineering Computations.
2. RAO G.S. Numerical Analysis
3. Grewal B S Numerical Methods In Engineering and Science
4. Das K K Advance Engineering Methods

Reference Books:

5. Rajaraman V Computer Oriented Numerical Methods
6. P. Kandasamy K. Thilagavathy. K Gumavathi, Numerical Methods & Chand & Company, 2nd Edition, Reprint 2012
7. S.S. Sastry, Introduction methods of Numerical Analysis, PHI, 4th Edition, 2005.
8. Erwin Kreyszig, Advanced Engineering Mathematics, 9th Edition, John Wiley & Son 2006

Course Outcomes:

After completion of this course students will be able to:

1. Find Numerical solution of various equations, which may be arising due to mathematical modelling based on engineering problems.
2. Demonstrate understanding of common numerical methods and how they are used to obtain approximate solutions to intractable mathematical problems.
3. Apply numerical methods to obtain approximate solutions to mathematical problems.
4. Analyse and evaluate the accuracy of common numerical methods.
5. Implement numerical methods in Matlab and Write efficient, well-documented Matlab code and present numerical results in an informative way.

Course Outcomes and their mapping with Programme Outcomes:

CO	PO												PSO		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	2	1	1							2	1	2	1
CO2	3	2	2	2	1							2	3	2	2
CO3	3	3	2	3	2							2	1	2	1
CO4	3	3	3	3	3							2	3	3	2
CO5	3	3	3	3	3							2	3	3	2

Weightage: 1-Sightly, 2-Moderately, 3-Strongly

SUB CODE	L	T	P	DURATION	IA	ESE	CREDITS
IT203TPC03	3	1	0	3 HOURS	30	70	4

Object Oriented Programming

Course Objectives:

1. To understand and Practice Programming Construct: Variable, Operators, Control Structures, Loop, Functions, learn the concept of class and object and develop classes for simple applications with C++.
2. To learn how to implement Constructors, copy constructors and destructor functions.
3. To learn how to overload functions and operators in C++.
4. To learn how to design C++ classes for code reuse and perform inheritance.
5. To learn working with files and handle exceptions in program.

UNIT I

Overview of C++ : Object oriented programming, Concepts, Advantages, Usage. C++ Environment: Program development environment, the language and the C++ language standards. Introduction to various C++ compilers, C++ standard libraries, Prototype of main() function, Data types. C++ as a superset of C, New style comments, main function in C++, meaning of empty argument list, function prototyping, default arguments and argument matching.
User defined data types: enumerated types, use of tag names, anonymous unions, scope of tag names Classes & Objects : Classes, Structure & Classes, Inline Function, Scope Resolution operator, Static Class Members: Static Data Member, Static Member Function, Passing Objects to Function, Returning Objects, Object Assignment. Friend Function, Friend Classes

UNIT II

Array, Pointers References & The Dynamic Allocation Operators: Array of Objects, Pointers to Object, Type Checking C++ Pointers, The This Pointer, Pointer to Derived Types, Pointer to Class Members, References: Reference Parameter, call by reference and return by reference Passing References to Objects, Returning Reference, Independent Reference, C++'S Dynamic Allocation Operators, Initializing Allocated Memory, Allocating Array, Allocating Objects.
Constructor & Destructor: Introduction, Constructor, access specifies for constructors, and instantiation, Parameterized Constructor, Multiple Constructor in A Class, Constructor with Default Argument, Copy Constructor, Destructor.

UNIT III

Overloading as polymorphism: Function & Operator Overloading : Function Overloading, Overloading Constructor Function Finding the Address of an Overloaded Function, Operator Overloading: Creating A Member Operator Function, Creating Prefix & Postfix Forms of the Increment & Decrement Operation, Overloading The Shorthand Operation (i.e., +=, -= etc), Operator Overloading Restrictions, Operator Overloading Using Friend Function, Overloading Some Special Operators like [ ], ( ), -, Comma Operator, Overloading << etc.

UNIT IV

Inheritance : Base Class Access Control, Inheritance & Protected Members, Protected Base Class Inheritance, Inheriting Multiple Base Classes, Constructors, Destructors & Inheritance, When Constructor & Destructor Function are Executed, Passing Parameters to Base Class Constructors, Granting Access, Virtual Base Classes .
Virtual Functions & Polymorphism: Virtual Function, Pure Virtual Functions, Early Vs. Late Binding.

UNIT V

Working with files: File & stream, Opening and closing a file, read ( ) and write ( ) functions, detecting end of file.

Templates and Exception Handling: Exception handling in C++, try, throw, catch sequence, multiple catch blocks, uncaught exceptions, catch-all exception handler

Course Outcomes:-

1. Understand the C++ language features. Use the control structure and data types in C++. Write simple programs using classes and objects.
2. Understand the concepts of arrays, pointers, references and use of dynamic allocation operators. Write simple programs to implement Constructor & destructor concepts.
3. Understand the concept of Operator overloading and type conversion.
4. Understand the concepts of inheritance and virtual functions.
5. Understand file handling concepts, generic class and I/O exception handling.

CO	PO												PSO		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	3	3	3	3								3	3	1
CO2	3	3	3	3	3								3	3	1
CO3	3	3	3	3	2								3	3	1
CO4	3	2	2	2	1								3	2	1
CO5	3	2	2	2	1								3	2	1

Weightage: 1-Sightly, 2-Moderately, 3-Strongly

Reference Books:

Object Oriented Programming with C++ by M. P. Bhave, S. A. Patekar, Pearson Education

Object Oriented Programming With C++ by E. Balaguruswamy.

Object Oriented Programming in turbo C++ by Robert Lafore.

Programming with C++ by D. Ravichandan.

Programming with C++ (SOS) by Hubbard.

SUB CODE	L	T	P	DURATION	IA	ESE	CREDITS
IT203PES06			4	3 HOURS	30	20	2

ANALOG ELECTRONIC CIRCUITS LAB

Course Objective

1. Understand the circuit configurations and connectivity of Amplifiers.
2. Study of frequency response of Amplifiers.
3. To measure the frequency of Various Oscillator circuits.
4. Understand the feedback configurations of transistor circuits.
5. Understand the circuit configurations and connectivity of Power and Tuned Amplifiers for the analysis of electronic circuits.

List of Experiment

1. To analyze different configuration of BJT.
2. To analyze different configuration of FET.
3. To determine frequency response of R-C coupled amplifier.
4. To analyze Voltage Series Feedback Amplifier.
6. To analyze Voltage Shunt Feedback Amplifier.
7. To analyze Current Series Feedback Amplifier.
8. To analyze Current Shunt Feedback Amplifier.
9. To determine frequency of oscillation and gain of RC Phase shift Oscillator.
10. To determine frequency of oscillation and gain of Hartley Oscillator.
11. To determine frequency of oscillation and gain of Colpitt's Oscillator.
12. To determine frequency of oscillation and gain of Wien Bridge oscillator.
13. To analyze different types of Power amplifier.
14. To analyze different types of Tuned amplifier.

Course Outcome:

1. Design various amplifiers like CE, CC, common source amplifiers and implement them using hardware
2. Plot frequency response & analyze BJT(single stage and multistage) amplifiers by determining gain and bandwidth.
3. Design and implement oscillator circuits & measure frequency of oscillation.
4. Design and implement Feedback circuits & measure its input and output impedances
5. To verify practically Efficiency of different types of power amplifier and determine resonant frequency of tuned amplifiers

CO	PO												PSO		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	3	3	3	3							3	3	3	3
CO2	3	3	3	3	3							3	3	3	3
CO3	3	3	3	2	2							3	3	3	2
CO4	3	2	3	2	2							3	3	3	2
CO5	3	2	3	2	1							3	3	3	2

Weightage: 1-Sightly, 2-Moderately, 3-Strongly

SUB CODE	L	T	P	DURATION/WEEK	IA	ESE	CREDITS
IT203PPC01	0	0	4	4 HOURS	30	20	2

DATA STURCTURE LAB

Course Objective

1. Understand and remember algorithms and its analysis procedure.
2. Introduce the concept of data structures through ADT including List, Stack, Queues.
3. To design and implement various data structure algorithms.
4. To develop application using data structure algorithms.
5. Compute the complexity of various algorithms.

Course Outcome

At the end of this lab session, the student will

1. Design programs using a variety of data structures such as Stacks, Queues, Array, Binary Trees, and Linked List.
2. Analyze and implement various kinds of searching and sorting techniques.
3. Have practical knowledge on the applications of data structures.
4. Design and analyze complexity of different algorithms.
5. Design advance data structure using non linear data structure.

List of Practical's

1. Write a C program for declaration, assignment, and accessing the arrays elements.
2. Write a C Program to Find Average Marks obtained by a class of 30 Students in a Test.
3. Write a C program to perform Array Insertion Operation.
4. Write a C program to perform Array Deletion Operation.
5. Write a C program to implement Linear Search.
6. Write a C program to implement Binary Search.
7. Write a C program to implement Bubble Sort.
8. Write a C program to implement Merging operation.
9. Write a program in C to create and display Singly Linked List.
10. Write a program in C to create a singly linked list of n nodes and count the number of nodes.
11. Write a program in C to insert a new node at the beginning of a Singly Linked List.
12. Write a program in C to insert a new node at the end of a Singly Linked List.
13. Write a program in C to insert a new node after a given location of Singly Linked List.
14. Write a program in C to delete first node of Singly Linked List.
15. Write a program in C to delete the last node of Singly Linked List.
16. Write a program in C to delete a node from the middle of Singly Linked List.
17. Write a program in C to search an existing element in a singly linked list.
18. Write C programs to implement the stack push operation using an array.

19. Write C programs to implement the stack pop operation using an array.

References books:

1. Lipschutz, “Data Structures with C” Schaum’s Outline Series, TMH.
2. Horowitz and Sahani, “Fundamentals of data Structures”, Galgotia Publication Pvt. Ltd.
3. R. Kruse etal, “Data Structures and Program Design in C”, Pearson Education Asia.
4. A. M. Tenenbaum, “Data Structures using C & C++”, Prentice-Hall of India Pvt. Ltd.
5. K Loudon, “Mastering Algorithms with C”, Shroff Publisher & Distributors Pvt. Ltd.
6. Jean Paul Trembley and Paul G. Sorenson, “An Introduction to Data Structures with applications”, McGraw Hill.
7. G A V Pai, “Data Structures and Algorithms”, TMH.
8. G.S.Baluja, “Data Structures through C”, Dhanpat Rai & Co.
9. Yashavant Kanetkar, “Data Structure Through C”, BPB Publication.

CO	PO												PSO		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	1	3	3	2	2								3	3	3
CO2	2	3	1	2	3								3	3	3
CO3	3	3	3	3	3								3	3	3
CO4	2	3	2	3	3								3	3	3
CO5	3	2	1	2	3								3	3	3

Weightage: 1-Sightly, 2-Moderately, 3-Strongly

SUB CODE	L	T	P	DURATION	IA	ESE	CREDITS
IT203PPC02	0	0	4	4 HOURS	30	20	2

DIGITAL ELECTRONICS LAB

Course Objectives

The objectives are to study

1. To familiarize with different number systems and conversion from one to other
2. To provide basic fundamental concepts associated with digital logic and circuit diagrams
3. To introduced the laws involved in Boolean algebra
4. To get basic experimental experiences in constructing digital circuits, measuring the experimental data and analysis of the results.
5. To develop skills to design various combinational and sequential circuits using electronics devices.

Course Outcomes (COs)

After studying this course the students would gain enough knowledge.

1. To acquire knowledge about electronic components and hardware devices required for designing digital electronics circuits.
2. Foster ability to identify, analyze and design combinational circuits.
3. Foster ability to design various synchronous and asynchronous sequential circuits.
4. To acquire knowledge about internal circuitry and logic behind any digital system.
5. To develop skill to build, and troubleshoot digital circuits.

List of Experiments

1. To implement Logic gates using ICs (7400, 7402, 7404, 7408, 7410, 7411, 7420, 7427, 7432, 7486).
2. Implementation of Combinational Circuits.
3. To verify NAND and NOR gates are universal gates.
4. Implementation of Combinational Logic Design using 74** ICs.
5. Simplification of Boolean expression using Karnaugh Map Method.
6. To implement Adder and Subtractor circuits:- (Half and Full using simple gates and universal gates).

7. To verify the truth table of Binary (2 bit) to decimal decoder and octal to decimal decoder.

8. Functional table verification of Latches (i) SR-Latch with NOR Gates (ii) SR-Latch with NAND Gates.

9. Functional table verification of different counters.

10. Functional table verification of shift registers.

CO	PO												PSO		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	2	3	1	-	-	-	1	-	-	3	3	2	1
CO2	3	3	3	3	1	-	-	-	1	-	-	3	3	3	1
CO3	3	3	3	3	2	2	-	-	2	1	-	3	3	3	1
CO4	3	3	3	3	2	2	-	-	2	1	-	3	3	3	1
CO5	3	3	3	3	3	3	-	-	3	2	2	3	3	3	1

Weightage: 1-Sightly, 2-Moderately, 3-Strongly

SUB CODE	L	T	P	DURATION	IA	ESE	CREDITS
IT203PPC03			4	3 HOURS	30	20	2

Object Oriented Programming with C++ Lab

Course Objectives:

1. To understand and Practice Programming Construct: Variable, Operators, Control Structures, Loop, Functions, learn the concept of class and object and develop classes for simple applications with C++.
2. To learn how to implement Constructors, copy constructors and destructor functions.
3. To learn how to overload functions and operators in C++.
4. To learn how to design C++ classes for code reuse and perform inheritance.
5. To learn working with files and handle exceptions in program.

List of Experiments:-

1. Write a program to display message using cout statement.
2. Write a program to calculate average of five numbers given by user.
3. Write a program to calculate compound interest given P, R and T.
4. Write a program to calculate factorial of a given number.
5. Write a program to generate n numbers of fibonacii series. Value of n should be provided by user.
6. Write a function to calculate the power of a number raised to another number using function. Write appropriate main() function to read and display the result.
7. Write a function factorial to calculate the factorial of a number, write appropriate main function also.
8. Write a function swap to swap the value of two integer variables. Write appropriate main function for the program.
9. Write a function to perform sorting using bubble sort algorithm. Use arrays to store the list of numbers. Also write main() function to read contents and display output.
10. Write a program to perform overloading of area function.
11. Write a program with overloaded volume function. Use volume function to calculate the volume of a cube, cone, sphere etc.
12. Write a program to calculate simple interest. Use default argument for rate. Write main function to exhibit the use of default argument.
13. Write a program to show the use of return by reference.
14. Write a program with at least one function made as inline.
15. Create a structure data type with data items roll number, name, and total marks.Write main function to read data for two students and also display the stored data.
16. Create a class named 'Student' with a string variable 'name' and an integer variable 'roll no'. Assign the value of roll no as '2' and that of name as "John" by creating an object of the class Student.

17. Write a program to print the area of a rectangle by creating a class named 'Area' having two functions.First function named as ‘readData' takes the length and breadth of the rectangle as parameters and the second function named as ‘calculateArea' returns the area of the rectangle. Length and breadth of the rectangle are entered through keyboard.
18. Write a program that would print the information (name, year of joining, salary, address) of three employees by creating a class named 'Employee'. The output should be as follows:

Name	Year of joining	Address
Ramesh	1994	64-C New Delhi
Sam	2000	68-D Bilaspur
John	1999	26-B-Banglore

19. Define a class to represent a bank account. Include the following members:

    Data members:

    1. Name of the depositor.
    2. Account number.
    3. Type of account.
    4. Balance amount in the account.

    Member functions:

    1. To assign initial values.
    2. To deposit an amount.
    3. To withdraw an amount after checking the balance.
    4. To display the name and balance.

    Write a main program to test the program.

20. Define a class to represent a bank account (FOR 100 CUSTOMERS). Include the following members:

    Data members:

    1. Name of the depositor.
    2. Account number.
    3. Type of account.
    4. Balance amount in the account.

    Member functions:

    1. To assign initial values.
    2. To deposit an amount.
    3. To withdraw an amount after checking the balance.
    4. To display the name and balance.

    Write a main program to test the program.

21. Create two classes DM and DB which store the value of distances. DM stores distances in meters and centimeters and DB in feet and inches. Write a program that can read values for the class objects and add one object of DM with another object of DB. Use friend function to carry out the addition operation. The object that stores result may be a DM object or DB object depending on the units in which result is required.
22. Create a class time to store time in hours and minutes. Write a program that can read values for the class objects and add one object with another object storing the result in third object. Use

object as arguments to sum() function and object return type from sum() function to assign the sum to the third object.

23. Write a program to demonstrate the use of Static Keyword. Write a program to print the names of students by creating a Student class. If no name is passed while creating an object of the Student class, then the name should be "Unknown", otherwise the name should be equal to the String value passed while creating the object of the Student class. Use “new” operator to create an instance of “name” member of class.
24. Write a program to add two complex number using a friend function. Use appropriate constructor function to initialize the object.
25. Write a program to initialize an object with another using copy constructor.
26. Write a program to show the highest scorer in a test out of three students who appeared in a exam. Use this pointer to refer to objects.
27. Write a program to show the highest scorer in a test out of three students who appeared in a exam. Use this pointer to refer to objects. (USER INPUT)
28. Write a program to illustrate the creation and destruction of objects.
29. Write a program to illustrate pointer to member and pointer to object concepts of 00P.
30. Create two objects of a class with two integer type members. Compare the two operators have same member values. Using overloaded = operator for comparison.
31. Create a class FLOAT that contains one float data member. Overload all the four arithmetic operators so that they operate on the objects of FLOAT.
32. Write a program to overload << and >> operators to display time object.
33. Write a program to overload <=, >== and == operator to compare time objects.
34. Write a program to perform overloading of function call operator.
35. Write a program to perform overloading of subscripting operator.
36. Write a Program to Concatenate two Strings Using Operator Overloading(+).
37. Write a program to convert time type object to an Integer value and integer type value to object of time type. Use appropriate data type conversion functions to perform the required conversion.
38. Write a program for an inventory of product in store. Use appropriate data type conversion functions to perform the required conversion.
39. Define a class Student with data members as rollno and name. Derive a class Fees from student that has a data member fees and functions to submit fees and generate receipt.. Derive another class Result from Student and display the marks and grade obtained by the student.

Course Outcomes:-

1. Understand the C++ language features. Use the control structure and data types in C++. Write simple programs using classes and objects.
2. Understand the concepts of arrays, pointers, references and use of dynamic allocation operators. Write simple programs to implement Constructor & destructor concepts.
3. Understand the concept of Operator overloading and type conversion. Write simple programs using overloaded operators.
4. Understand the concepts of inheritance and virtual functions. Write simple programs to implement inheritance and virtual functions.

5. Understand file handling concepts, generic class and I/O exception handling. Write small programs to implement file handling concepts and exception handling.

CO	PO											PSO		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	2	1	3							2	3	2	3
CO2	3	2	2	1	3							2	3	2	3
CO3	3	2	2	1	3							2	3	2	3
CO4	3	2	2	1	3							2	3	2	3
CO5	3	2	2	1	3							2	3	2	3

Weightage: 1-Sightly, 2-Moderately, 3-Strongly