Course objectives:

To enable the students to obtain the knowledge of Engineering Mathematics in the following topics:

- Numerical methods to solve algebraic and Transcendental equations and Eigen values and Eigen vectors
- Interpolation methods and Numerical integration
- Fourier Series and Fourier transformation and Z transformation and its application in Engineering fields
- Partial Differential equations and its applications

Module I


10 hours

Module II


10 hours
Module III


**Calculus of variations**: Variation of functional, extremal of functional, variational problems, Euler's equations, standard variational problems including geodesics, minimal surface of revolution. Problems.

10 hours

Module IV

**Fourier series**: Periodic functions, Fourier series with periods \((0,2\pi),(-\pi,\pi),(0,2l)\) and \((-l,l)\). Half range fourier series. Practical harmonic analysis. Problems.

**Fourier Transform**: Finite and Infinite transforms. Fourier sine and cosine transforms, properties. Inverse transform. Problems.

11 hours

Module V

**Partial differential equations (PDE)**: Formation of PDE by eliminating arbitrary constants and arbitrary functions. Solution of non-homogeneous PDE by direct integration method. Solution of homogeneous PDE involving derivative with respective one independent variable only. Method of separation of variables. (First and second order equations). Solution of Langrange's linear PDE of the type \(Pp+Qq=R\). Problems.

**Applications of PDE**: Derivation of one dimensional wave and heat equations. Various possible solutions of wave equation, heat equation and Laplace equation by the method of separation of variables with given conditions. Problems.

11 hours

**Course Outcomes:**

After completion of this course, the students will be able to:

1. Solve the numerical problems in algebraic, transcendental equations, Eigen values and Eigen vectors. Computation of interpolation polynomials and numerical integration (C3)
2. Analyze discrete type system using convolution and the Z-transform (C4)
3. Determine Fourier transform for continuous time signals and systems (C4)
4. Construction of Fourier series for periodic signals and Fourier series to analyze circuits (C3)
5. Determine solution of wave, heat and Laplace equations (C4)

C3,C4: Cognitive levels.
**SEE Pattern:**
Question paper will have TEN questions.
Each full question consisting of 20 marks
There will be TWO full questions (with maximum of four sub-questions) covering all the entire topic under module.
The student shall answer FIVE full questions, selecting one full question from each module

**Text Books:**

**References Books:**
4. Introductory methods of numerical analysis by S.S.Sastry

**DISCRETE MATHEMATICS AND GRAPH THEORY**

<table>
<thead>
<tr>
<th>Sub Code</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>15IS32</td>
<td>03</td>
</tr>
<tr>
<td>CIE: 50 Marks</td>
<td>SEE: 50 Marks</td>
</tr>
<tr>
<td>Hrs/Week: 03 Hrs</td>
<td>Total Hrs: 42</td>
</tr>
<tr>
<td>SEE:03 Hrs.</td>
<td></td>
</tr>
</tbody>
</table>

**Module I**
**Set Theory:** Sets and Subsets, Set Operations and the Laws of Set Theory, Counting and Venn Diagrams, A First Word on Probability, Countable and Uncountable Sets. **Fundamentals of Logic:** Basic Connectives and Truth Tables, Logic Equivalence – The Laws of Logic, Logical Implication – Rules of Inference

**9 HOURS**

**Module II**
**Fundamentals of Logic contd.:** The Use of Quantifiers, Quantifiers, Definitions and the Proofs of Theorems, Mathematical Induction,
**Relations and Functions:** Cartesian Products and Relations, Functions – Plain and One-to-One, Onto Functions – Stirling Numbers of the Second Kind, Special Functions, The Pigeon-hole Principle

**8 HOURS**
Module III

Module IV
Introduction to Graph Theory: Definitions and Example, Sub graph, Complements and Graph Isomorphism, Vertex Degree, Euler Trails and Circuits. Planar graphs, Hamilton Paths and Cycles, Graph coloring, Chromatic Polynomials, Graph colouring, chromatic Polynomials;

Module V
Trees: Definitions, Properties, and Examples, Routed Trees, Trees and Sorting, Weighted Trees and Prefix Codes

Course Outcomes:
After completion of this course, the students will be able to:
1. Illustrate the notion of mathematical thinking, mathematical proofs fundamentals of set theory, functional, relational, Boolean properties and operations able to apply them in problem solving. (C2,C1)
2. Solve problems which involve discrete data structures such as relations and discrete functions (C2,C3)
3. Use effectively algebraic techniques to analyze and apply basic discrete structure and theory of combinatorics. (C3,C4)
4. Demonstrate knowledge of fundamental concepts in graphs, trees and its properties using various modeling techniques. (C1,C3)

SEE Pattern:
1. The question paper will have TEN questions.
2. There will be TWO questions in each module, covering all the topics.
3. The student need to answer FIVE full questions, selecting ONE full question from each module.
Text Books:
2. "Introduction To Graph Theory" By Gary Chartrand, Ping Zhang, TMH, 2006

References Books:
4. Discrete Mathematical structures, Dr. D. S. Chandra Shekar, Prism.

LOGIC DESIGN

Sub Code: 15IS33
Credits: 04
CIE: 50 Marks
SEE: 50 Marks
Hrs/Week: 3 Hrs (Theory) + 2hrs. (Tutorial)
Total Hrs: 52
SEE: 03 Hrs.

Module I

10 HOURS

Module II
Data-Processing Circuits: Multiplexers, Demultiplexers, 1-of-16 Decoder, BCD to Decimal Decoders, Seven Segment Decoders, Encoders, Exclusive-OR Gates, Parity Generators and Checkers, Magnitude Comparator, Programmable Array Logic, Programmable Logic, HDL Implementation of Data Processing Circuits. Arithmetic Building Blocks, Arithmetic Logic Unit FlipFlops: RS Flip-Flops, Gated Flip-Flops, Edge-triggered RS FLIP-FLOP.
Module III
Clocks, Flip-Flops: Clock Waveforms, TTL Clock, Schmitt Trigger, Clocked D FLIP-FLOP, Edge-triggered D FLIP-FLOP, Edge-triggered JK FLIP-FLOP, FLIP-FLOP Timing, JK Master-slave FLIP-FLOP, Switch Contact Bounce Circuits, Various Representation of FLIP-FLOPs, Analysis of Sequential Circuits, HDL Implementation of FLIP-FLOP

Module IV
Registers: Types of Registers, Serial In - Serial Out, Serial In - Parallel out, Parallel In - Serial Out, Parallel In - Parallel Out, Universal Shift Register, Applications of Shift Registers, Register implementation in HDL.
Counters: Asynchronous Counters, Decoding Gates, Synchronous Counters, Changing the Counter Modulus. Decade Counters, Pre settable Counters, Counter Design as a Synthesis problem, A Digital Clock, Counter Design using HDL

Module V

Course Outcomes:
After completion of this course, the students will be able to:
1. Design of logical circuit using K-map, SOP, POS concepts (C5,C1)
2. design and develop encoder, coders and multiplexer using combinational circuits and also Design and implement programmable logical arrays (C5)
3. Understand the design concepts of different types of combinational and sequential circuits (C2)
4. Evaluate A/D and D/A conversions (C6)
SEE Pattern:
1. The question paper will have TEN questions.
2. There will be TWO questions in each module, covering all the topics.
3. The student needs to answer FIVE full questions, selecting ONE full question from each module.


Reference Books:

DATA STRUCTURES WITH C

Sub Code: 15IS34
Credit: 04
CIE: 50 Marks
SEE: 50 Marks
Hrs/Week: 3 Hrs (Theory) + 2 hrs (Tutorial)
Total Hrs: 52
SEE: 03 Hrs.

Module I

C Language Features:

Structures and Unions: Structure definition, Giving values to members, Structure initialization, Comparison of structure variables, Arrays of structures, Arrays within structures, Structures within structures, Structures and Functions, Unions, Size of structures, Bit fields.

Pointers: Understanding pointers and the address of operator, Declaring and initializing pointer, Accessing a variable through its pointer, Pointers and arrays, Pointers and character strings, Pointers and Functions, Pointers and Structures.

Dynamic Memory Allocation: Meaning of dynamic memory allocation, malloc, calloc, free and realloc functions, Pointers revisited.
File Management: Defining and opening a file, Closing a file, I/O operations on files, Error handling during file operations, Random access to files, Command line arguments.

10 HOURS

Module II
The Stack: Definition and examples: Primitive operation, Example. Representing stacks in C: Implementing the pop operation, Testing for exceptional conditions, Implementing the push operation.
Example: Infix, postfix and prefix, Basic definitions and examples, Evaluating a postfix expression, Program to evaluate a postfix expression, Converting an expression from infix to postfix, Program to convert an expression from infix to postfix.

10 HOURS

Module III
Queues and Lists: The queue and its sequential representation: C implementation of queues, Insert operation, Priority queue, Array implementation of a priority.
Linked Lists: Inserting and removing nodes from a list, Linked implementation of stacks, getnode and freenode operation, Linked implementation of queues, Linked list as a data structure, Example of list operations, Header nodes.
Queues and lists Contd.: Lists in C: Array implementation of lists, Limitations of array implementation, Allocating and freeing dynamic variables, Linked lists using dynamic variable, Queues as lists in C, Examples of list operations in C.

11 HOURS

Module IV
Trees: Binary trees: Operations on binary trees, Applications of binary trees.
Binary tree representation: Node representation of binary tree, Internal and external nodes, Implicit array representation of binary trees, Choosing a binary tree representation, Binary tree traversal. Trees and their application: C representation of trees, Tree traversals, General expressions as trees, Evaluating an expression tree, Constructing a tree. Examples

11 HOURS
Module V

**Sorting:** Binary tree sort, Simple insertion sort, Shell sort, Address calculation sort, Radix sort.

**Searching:** Sequential searching, Searching an ordered table, Indexed sequential search, Interpolation search. Tree searching: Inserting into a binary search tree, Deleting from a binary search tree. Hashing: Resolving hash clashes by open addressing, Choosing a hash function.

10 HOURS

**Course Outcomes:**

After completion of this course, the students will be able to:

1. Describe the concept of dynamic memory management data types, array data structure and asymptotic notations. (C1,C2)

2. Choose appropriate data structures as applied to specified problem definition and implement them using C programming language. (C3,C6)

3. Compare, analyze and implement different sorting techniques. (C6,C2,C4)

4. Implement and explain the standard structure of tree, Binary trees and binary search trees. (C6,C2)

**SEE Pattern:**

1. The question paper will have TEN questions.
2. There will be TWO questions in each module, covering all the topics.
3. The student need to answer FIVE full questions, selecting ONE full question from each module.

**Text Books:**

1. Data structures using C and C++ by Yedidyah Langsam and Moshe J. Augenstein and Aaron M Tenenbaum, PHI/Pearson
2. “C” Complete Reference by Herbert Sheldit.

**Reference Books:**

1. Robert Kruse, C.L Tondo and Bruce Leung, Data structures & Program design in C, II edition, Pearson education, Asia
Module I
Machine Instructions and Programs: Memory Location and Addresses, Memory Operations, Instructions and Instruction Sequencing, Addressing Modes, Assembly Language, Basic Input and Output Operations, Stacks and Queues, Subroutines, Additional Instructions, Encoding of Machine Instructions

9 HOURS

Module II

8 HOURS

Module III

8 HOURS

Module IV

8 HOURS

Module V

9 HOURS
Course outcomes:
After completion of this course, the students will be able to:
1. Identify the operations concepts structures and performances measures of computers with machine instructions and programs. (C1,C4)
2. Discuss various methods of handling of interrupts , multiple devices and interface circuits (C1,C3)
3. Describe different storage devices, memory management and virtual memory concepts.(C2)
4. Identify and apply instruction execution methods using hardwired and micro program control techniques and the embedded system concepts .(C3,C1)

SEE Pattern :
1. The question paper will have TEN questions.
2. There will be TWO questions in each module, covering all the topics.
3. The student need to answer FIVE full questions, selecting ONE full question from each module.

Text Book:

Reference Book:

OBJECT ORIENTED PROGRAMMING WITH C++ & JAVA

Subject Code: 15IS36
Hours/Week: 3 Hrs. (Theory) + 2 Hrs. (Tutorials)  Credits: 04
CIE: 50 Marks  S.E.E: 50 Marks
SEE: 03 Hrs.  Total Hrs: 52

Module I


Class and Objects: Introduction to Classes and Objects. Member Functions and Member Data, Objects and Functions, Objects and Arrays, Namespaces, Nested Classes.
Constructors and Destructors: Constructors, Destructors, Copy constructors.

11 Hours

Module II
Inheritance: Introduction to Inheritance, Base Class and Derived class, Pointers, Function Overriding, Base Class Initialization. The Protected Access Specifier, Deriving by Different Access Specifiers, Different kinds of inheritance.
Virtual functions and dynamic polymorphism: the need for virtual functions, virtual functions, the mechanism of virtual functions, pure virtual functions,
Templates: Introduction, function templates, class templates.

10 Hours

Module III
Introduction to JAVA: Overview of JAVA, Java applications, JDK, Compiling Java Program, Java Interpreter, Byte code, JVM, Simple JAVA Programs. Primitive, non-primitive data types, Type casting, Arrays and strings.
Operators & Expressions: Arithmetic operators, Bitwise operators, Relational Operators, Logical Operators, The Assignment Operators, The ?: operators, Operator precedence; Logical expression; Control statements, Selection statements, Iteration statements, Jump statements.
Class, objects, Methods and Inheritance: Classes in Java, Class fundamentals, Super classes, Constructors; Creating instances of class; Methods; Method overloading;

10 Hours

Module IV
Inheritance: Simple, Multiple and multilevel inheritance, overriding, overloading, using abstract classes, using final with inheritance.
Exception Handling: Exception type, Multiple catch statements, uncaught exceptions, using try and catch block, Nested try statements, Multiple catch statements Java built in exceptions.
Event Handling: Event handling mechanisms, The delegation event model, event classes, source of events, Event listener interfaces, Adapter classes, inner classes.

11 Hours

Module V
Multithread Programming: Java thread model, thread priorities, Synchronization, Messaging, thread class and runnable interface, main thread, creating a thread, multiple threads, stopping and blocking a thread, Thread life cycle, thread methods, thread exceptions.
Applet Programming: Applet class, Applet basics, Two types of applets, Applet architecture, Applet skeleton, Applet lifecycle, Simple applet display method,
Managing I/O Files in JAVA: Stream classes, byte stream classes, character stream classes, other I/O classes, I/O exceptions, Reading writing character, Reading writing bytes.

10 Hours
Course outcomes:

After completion of this course, the students will be able to:
1. Describe the need of using object oriented programming in the real world application (C1,C2)
2. Describe the OOP Principles, objectives and design strategies that ease the real time challenges.(C3,C4)
3. Design and Develop the application programs using class, objects and object oriented features (C5)
4. implement and evaluate for the sustainability of the object oriented features in real time application.(C6,C5)

SEE Pattern:

1. The question paper will have TEN questions.
2. There will be TWO questions in each module, covering all the topics.
3. The student need to answer FIVE full questions, selecting ONE full question from each module.

Text Books:

   (Chapter 1-10).

Reference Books:

1. Given any 4-variable logic expression, simplify using Entered Variable Map and realize
the simplified logic using 8:1 multiplexer IC.
2. Realize a full adder using 3-to-8 decoder IC and 4 input NAND gates.
3. Realize a J-K Master/Slave Flip-Flop using NAND gates and verify its truth table
4. Design and implement a mod-n(n<8)synchronous up counter using J-K Flip-Flop
   ICs.
5. Design and implement a ring counter using 4-bit shift register.
6. Design and implement an asynchronous counter using decade counter IC to count
   up from 0 to n(n<=9).
7. Design a 4-bit R-2R ladder D/A converter using Op-Amp. Determine its accuracy and
   resolution.
8. Design Schmitt trigger circuits using Op-Amp for given UTP and LTP values
9. Using 555 timer to design the following:
   a) Astable Multivibrator for a given frequency and duty cycle
   b) Monostable Multivibrator for a given pulse width tp
10. Implement shift registers for following operations
    i) Serial-in serial out.
    ii) Parallel in serial out.

Course Outcomes:
After completion of this course, the students will be able to:
1. Design of logical circuits using K-Map concepts.(C5)
2. Evaluate Flip-Flops, Counters and Shift Registers.(C6)
3. Design various Op-amp Circuits(C5)
4. Design code converter circuits and multi vibrator circuits(C5)

Note: For SEE, students will be asked to do one program which may be related to
the above list of programs.
1. Write a C program to create a sequential file with at least five records. Each record having the structure shown below.

<table>
<thead>
<tr>
<th>Name</th>
<th>Marks1</th>
<th>Marks2</th>
<th>Marks3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-Zero positive integer</td>
<td>25-characters</td>
<td>Positive integer</td>
<td>Positive integer</td>
</tr>
</tbody>
</table>

Write necessary function
a) To display all the records in the file
b) To search for a specific record based on the USN. In case the record is not found. Suitable message should be displayed. Both the options in this case must be demonstrated.

2. Write a C program, which accepts the Internet Protocol(IP) address in decimal dot format (ex. 153.18.8.105) and converts it into 32-bit long integer (ex. 2568095849) using strtok library function and unions.

3. Write a program using Recursion i) Solving the Towers Of Hanoi problem ii) Searching an element on a given list of integers using the Binary search method.

4. Write a program to perform push, pop and display operations on a stack using i) Linear array ii) pointers.
The program should print appropriate messages for stack overflow, stack underflow and stack empty.

5. Write a C program to convert and print a given valid parenthesized infix arithmetic expression to a postfix expression. The expression consists of single character operands and the binary operators +(plus), -(minus), *(multiply) and /(divide).

6. Write a C program to evaluate a valid suffix/postfix expression using stack. Assume that the suffix/postfix expression is read as a single line consisting of non-negative single digit operands and binary arithmetic operators. The arithmetic operators are +(plus), -(minus), *(multiply) and /(divide).
7. Write a program to perform insert, delete and display operations on a queue using
i) Linear array ii) pointers.
The program should print appropriate messages for queue full and queue empty.

8. Write a C program to simulate the working of a circular queue of integers using an array.
Provide the following operations
   a) Insert
   b) Delete
   c) Display

9. Write a C program using dynamic variables and pointers, to construct a singly linked list
   consisting of the following information in each node: student id(integer), student
   name(character string) and semester(integer). The operations to be supposed are:
   a) The insertion operation
      i) At the front of a list
      ii) At the back of the list
      iii) At any position in the list
   b) Deleting a node based on student id. If the specified node is not present in the list an
      error message should be displayed. Both the operations should be demonstrated.
   c) Searching a node based on student id and update the information content. If the
      specified node is not present in the list an error message should be displayed. Both
      situations should be displayed.
   d) Displaying all the nodes in the list.

10. Write a C program to support the following operations on a doubly linked list where
    each node consists of integers:
      a) Create a doubly linked list by adding each node at the front
      b) Insert a new node to the left of the node whose key value is read as an input.
      c) Delete the node of a given data, if it is found, otherwise display appropriate message
      d) Display the contents of the list.

11. Write a C program using dynamic variables and pointers to construct a stack of integers
    using singly linked list and to perform the following operations.
      a) PUSH
      b) POP
      c) Display
    The program should print appropriate messages for stack overflow, and stack empty.
12. Write a C program using dynamic variables and pointers to construct a queue of integers using singly linked list and to perform the following operations.
   a) Insert
   b) Delete
   c) Display
   The program should print appropriate messages for queue full and queue empty.

13. Write a C program
   a) To construct a binary search tree of integers.
   b) To traverse the tree using all the methods i.e., inorder, preorder and postorder.
   c) To display the elements in the tree.

14. Write a C program to evaluate an expression tree.

Course Outcomes:
After completion of this course, the students will be able to:
1. Design and implement C program for implementing stacks, queues, linked lists.(C3,C5)
2. Apply efficient design methods for program development.(C3)
3. Apply the different data structures for implementing solutions to practical problems.(C3)
4. Develop searching and sorting programs.(C5).
Note: For SEE, students will be asked to do programs which may be related to the above list of programs.

OBJECT ORIENTED PROGRAMMING LAB USING C++ & JAVA

Subject Code: 15IS39 Credits: 01
Hours/Week: 02 Hrs(Practical) SEE: 50 Marks
CIE: 50 Marks SEE:03 Hrs.

Practice Programs using C++ and JAVA
1. Write a program to find the area and perimeter of rectangle using class and objects using.
2. Write a program to find the area of square, rectangle using function overloading.
3. Write a simple program to derive new classes using multiple inheritance.
4. Write a simple program to derive new classes using multilevel inheritance.
5. Write a applet program for menu demonstration, menu bar should contain File, Edit, View and its submenus.
6. Write a program to print the following outputs using for loops
C++ Lab Programs:

1) Create a simple class **STUDENT** containing data members roll no, name age & display the contents using setdata() and Outdata() methods.
   Test the program with
   a) Member function inside the body of the student class.
   b) Member function outside the body of the student class (using :: operator).

2) Write a C++ program to create a class **ACC** with data members, accno, balance. Create objects ACC1, ACC2 and ACC3. Write a member function to transfer money from ACC3 to ACC1. Display the balance in all accounts.

3) Create a class called **QUEUE** perform insertion and deletion of elements from the queue using constructors and destructors.

4) Write a C++ program to create a class called **STACK** using an array of integers. Implement the following operations by overloading + & -.
   i) s1=s1+ element; where s1 is an object of the class STACK and element is an integer to be pushed on to top of the stack.
   ii) s1=s1-; where s1 is an object of the class STACK and – operator pops the element. Handle the STACK Empty and STACK Full conditions. Also display the contents of the stack after each operation, by overloading the operator <<.

5) Write a C++ program to create a class **NAME** and implement the following operations. Display the result after every operation by overloading the <<.
   i) NAME firstname = “Herbert”
   ii) NAME lastname = “Schield”
   iii) NAME fullname = firstname +lastname
       (Use copy constructor)

6) Write a C++ program to exchange two numbers using function overloading.

7) Design three classes called **STUDENT, EXAM and RESULT**. The student class has data members Such as those represent Rollno, Name and Branch etc. Create the class EXAM by inheriting the STUDENT class. The EXAM class adds data members
representing the marks scored in six subjects. Derive the RESULT class from the EXAM class and it has its own data members. Such as total_marks. Write an interactive program to model this inheritance relationship.

8) Create classes RESERVATION, ADULT, SENIOR_CITIZEN, CHILD. The Reservation class containing data members, Name of passenger, age, date of journey, Source, Destination, Ticket charge. Write an interactive program to display the ticket charges depending upon the category of passenger.

The classes ADULT, SENIOR_CITIZEN, CHILD are the derived class of RESERVATION.

(Note: Category CHILDREN = ½ of adult ticket charge. Senior_citizen = ¼ of adult ticket charge.)

JAVA Programs:

1. The numbers in the sequence 1 1 2 3 5 8 13 21 are called Fibonacci numbers. Write a program using do-while loop to calculate and print the first m Fibonacci numbers. (Hint: After the first two numbers in the series, each number is the sum of the two preceding numbers).

2. Write a program which will read a text and count all occurrences of a particular word

3. Write a Java program to create class ACC with data members, accno, balance. Create objects ACC1, ACC2 & ACC3. Write a member function to transfer money from ACC3 to ACC1, display the balance in all accounts.

4. Write a Java program to implement the concept of multiple inheritance using interfaces.

5. Write a Java program for handling mouse events.

6. Develop an applet that receives an integer in one text field, and computes its factorial value and returns it in another text field, when button name “compute” is clicked.

7. Write a Java program that creates three threads. First thread displays “Good Morning” every one second, the second thread displays “hello” every two seconds and the third thread displays “Welcome” every three seconds.

8. Write a java program to store and retrieve integers using data streams on a single file.

Course outcomes

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

1. Describe the fundamentals of object oriented programming. (C1, C2)
2. Describe the programming principles in designing applications using C++ and JAVA (C3, C4)
3. Design and Develop the modules for a given problem description
4. Evaluate programming features for reusability of code using inheritance. (C6, C5)

SEE Pattern:

1. Practice programs will not be asked in the exam.
2. Student is asked to execute two programs selecting one from C++ and JAVA in SEE.