# COMPUTER SCIENCE(H-4705) SYLLABUS STD XII

### ACADEMIC YEAR : 2024-2025

| BOARD EXAM ( MARKS:- 70) | | |
|---|---|---|
| **UNIT NO** | **NAME OF UNIT** | **MAX MARKS** |
| I | PROGRAMMING IN C++ | 20 |
| II | DATA STRUCTURE IN C++ | 20 |
| III | FILE HANDLING IN C++ | 10 |
| IV | BOOLEAN ALGEBRA | 10 |
| V | COMPUTER NETWORK | 10 |
| | TOTAL | 70 |

| MID-TERM ( MARKS-20) | | |
|---|---|---|
| **UNIT NO** | **NAME OF UNIT** | **MAX MARKS** |
| I | PROGRAMMING IN C++ (Sub Units:-01 to 09) | 10 |
| IV | BOOLEAN ALGEBRA | 10 |
| | TOTAL | 20 |

| FIRST TERM ( MARKS-60) | | |
|---|---|---|
| **UNIT NO** | **NAME OF UNIT** | **MAX MARKS** |
| I | PROGRAMMING IN C++ (Sub Units:-1 to 10) | 25 |
| IV | BOOLEAN ALGEBRA | 10 |
| III | FILE HANDLING IN C++ | 10 |
| II | DATA STRUCTURE IN C++ (Sub Units:-1 & 2) | 15 |
| | TOTAL | 60 |

## INTERNAL ASSESSMENT (Marks : 20 )

1) **Practical Exam (10 marks)**

   **Only 1 program from the given list of 10 short programs should be tested on the following parameters.**

   | | |
   |---|---|
   | **Program Design** | **– 3 marks** |
   | **Logical clarity** | **– 4 marks** |
   | **Input** | **– 1 mark** |
   | **Correct Output** | **– 2 marks** |

2) **Assignment (10 marks)**
   **SUGGESTED ASSIGNMENTS (any one)**

   1) Program(one) based on unit 1

   2) Program(one) based on unit II

   3) Program(one) based on unit III

   **Assignment to be evaluated based on the following parameters.**

   | | |
   |---|---|
   | **Program Design** | **– 3 marks** |
   | **Logical clarity** | **– 4 marks** |
   | **Input** | **– 1 mark** |
   | **Correct Output** | **– 2 marks** |

## FIRST-TERM PRACTICAL EXAM (20 MARKS)

- Any one program to be assessed from list of long programs from seriel number 1 to 7

  Program should be  tested on the following parameters.

  | | |
  |---|---|
  | Program Design | – 6 marks |
  | Logical clarity | – 7 marks |
  | Input | – 1 mark |
  | Correct Output | – 2 marks |
  | Journal | - 2 marks |
  | Viva | - 2 marks |

## BOARD PRACTICAL EXAM (20 MARKS)

- Any one program to be assessed from list of long programs from seriel number 1 to 10

  Program should be  tested on the following parameters.

  | | |
  |---|---|
  | Program | – 10 marks |
  | Journal | - 5 marks |
  | Viva | - 5 marks |

**PROGRAMMING IN C++ (20MARKS)**
**1. C++ Fundamentals:**
  ➢ C++ character set
  ➢ Identifiers and Keywords
  ➢ Data Types: int, float, char, double, void
  ➢ Qualifiers: short, long, signed, unsigned, const
  ➢ Constants (Integer, Floating point, character, string, enumeration constants, symbolic constants
  ➢ Escape Sequence (\b, \t, \n, \v, \a, \f,\r, \0)
  ➢ Variables and Declaration, Dynamic initialization of variables, reference Variables
**2. Operators and Expressions:**
  ➢ Unary Operators: unary minus,++, - -, !, sizeof(),typecast
  ➢ Arithmetic Operators: *, /, %,+,-
  ➢ Relational Operators: <, <=, >, >=
  ➢ Equality Operators: ==, !=
  ➢ Logical Operators: &&, ||
  ➢ Conditional Operator: ?:
  ➢ Assignment Operator: =, +=, -=, *=, /=, %=
  ➢ Scope resolution operators(::)
  ➢ Memory management operators: new(), delete()
  ➢ Operator precedence and associativity
**3. Data Input and Output**
  ➢ Header file <iostream.h>
  ➢ Using cin and cout with Insertion and extraction operators
  ➢ Manipulators: Definition, Header file <iomanip.h>
     setw, endl, setprecision, setfill, setiosflags, resetiosflags
     Flags : ios::left, ios::right, ios::scientific, ios::fixed,
     ios::showpos, ios::showpoint, ios::skipus, ios::unitbuf.
**4. Use of editor,** basic commands of editor, Compilation, Linking and
     Execution of Program, Debugging.
**5. Control Statements:**
     If-else statement, while statement, do-while statement, for statement,
     switch statement. break statement, continue statement Comma operator.
**6. Functions:**
     Definition, Concept, General Form, Function Declaration, Function Definition,Scope of a variable, Local and Global variables
     Function Call (pass by value, pass by reference using pointers and pass by reference using reference variable) Calling Function with arrays as parameters, Return by Reference, Inline Functions,
     Functions with Default Parameters, Function overloading
     Built-in Functions:
     <string.h>: strlen(), strcmp(), strcat(), strcpy()
     <math.h> : log(), log10(), pow(), sqrt(), sin(), cos(), abs()

&lt;ctype.h&gt;: isalnum(), isdigit(), islower(), isupper(), tolower(),
                toupper(), isalpha(),isspace()
&lt;stdio.h&gt;: gets(),puts(),getchar(),putchar()
&lt;conio.h&gt;: clrscr(),getch()

## 7. Basic concepts of Object Oriented Programming:

Definition, Objects, Classes, Data Abstraction, Data Encapsulation, Inheritance, Polymorphism Characteristics of Object Oriented Programming

## 8. Classes and Objects:

Definition of class and object, Declaration of class, Defining member functions (inside the class and outside the class), Creating Objects, Accessing Class Members, Array of objects, Objects as function argument, Functions returning objects.

## 9. Constructors and Destructors

Definition and characteristics of constructors, Default Constructor, Constructor with Default Arguments, Parameterized constructors (explicit call, implicit call), Constructer overloading, Copy Constructor, Dynamic Constructor, Dynamic Initialization of objects, Destructor Definition and characteristics.

## 10. Inheritance:

Definition, Concept of Inheritance: Base , and derived classes,
Abstract classes Type of Inheritance: single, multiple, multilevel, hierarchical and hybrid. Defining Derived Class: Visibility modes (public, private, protected) Public Derivation, Private Derivation, Protected Derivation, Constructors in Derived classes, Containership Virtual base classes.


## UNIT II :
## DATA STRUCTURES USING C++        (20 MARKS)
## 1) Arrays:-

➢ One Dimensional arrays Definition, Declaration, Reading, Displaying, Accessing. Algorithm and Program for inserting and deleting an element in an array. Memory allocation. Sorting: Definition, sorting techniques (Insertion Sort, Selection Sort, Bubble Sort algorithms and programs) Searching: Definition, Searching Techniques (Linear Search, Binary Search Algorithms and Programs) Merging: Definition, Algorithm and Program to merge two sorted arrays Applications: Insertion of an element in a sorted array, Displaying common elements of two single dimensional arrays and other applications. •
➢ Two Dimensional Arrays Definition, Declaration, Reading, Displaying, Accessing. Applications: Matrix Addition, Transpose of a matrix, Matrix Multiplication, Representation of sparse matrix in 3- Tuple form, and other applications.

## 2) Structures and Pointers:

➢ Pointers- Definition, Concept, Declaration, Pointer to one and two dimensional array.
➢ Structures- Definition, Concept, Declaration, Structure variable, array of structures, pointer to a structure variable, pointer to array of structures.

## 3) Linked lists

➢ Singly Linked list: Concept, Definition, Diagram, Operations (Creation, Display, Deletion of a node at any position, Insertion of a node at any position).
➢ Circular Linked list: Concept, Definition, Diagram.
➢ Doubly Linked List: Concept, Definition, Diagram
➢ Applications of singly linked list- Linear search and other applications.
➢ Stacks: Concept, Definition, and implementation of a stack using linked list(PUSH,POP and display). Applications of stacks (Infix, Postfix and Prefix Notations of expressions,

Conversion of infix to postfix using stacks [Algorithms and problems only], Evaluation of postfix expressions[Algorithms and problems only]).

➢ Queues: Linear queue- Concept, Definition, Implementation of queue using linked list( Add, Delete and Display). Circular Queue- Concept, Definition.

# UNIT III:
## FILE HANDELING (10 MARKS)

➢ Files: Definition, Types of files-Text and Binary.
➢ Stream Classes and their Member Functions. Ifstream- get(), getline(), read(), seekg(),tellg(), Open(),close(),eof() Ofstream- put(), seekp(), tellp(), write(), Open(),close(). Fstream.
➢ File Modes-ios:: app, ios::ate, ios::in, ios::out, ios::binary, ios::trunc, ios::nocreate, ios::noreplace
➢ Opening a file using constructor and using open member function.
➢ Closing a file.
➢ Detecting the end of a file.
➢ File Pointer and their manipulation.
➢ Text Files: Creation, Display and File Processing (Character and String based processing)
➢ Binary Files: Creation, Display and File Processing (Appending, Inserting, Deleting, Updating, Searching, Splitting and Merging)

# UNIT IV:
## BOOLEAN ALGEBRA (10 marks)

➢ Basics of Boolean Algebra :- Evolution of Boolean Algebra , Basic Terminology – Logical Statements, Logical Constants, Binary Valued Quantities , Compound Statements, Truth Table.
➢ Logical operators:- NOT, AND, OR , Switch, Switching Circuits (NOT, AND, OR).
➢ Postulates of Boolean algebra:- Closure Property, Commutative Property , Associative Property Distributive Property , Identity Property , Inverse Property .
➢ Laws of Boolean Algebra : Idempotent Law, Distributive Law, Absorption Law, Involution law.
➢ DeMorgan's Law and their applications.
➢ Principle of Duality in Boolean algebra.
➢ Derivation of Boolean expression:
➢ Minterm , Maxterm , Shorthand Notation, Canonical Form, Sum of Product form (SOP), Product of Sum form (POS), Conversion of SOP to POS and vice versa, simplification of boolean expressions using postulates and laws of Boolean Algebra.
➢ Karnaugh Maps: Two variable K map, Three variable K map, Four variable K map, Pairing , Quads, Octet in K map, Simplification of K maps up to four variables , Overlapping groups , map rolling, eliminating redundant groups, use of K map for simplification and conversion of Boolean expression.
➢ Logic gates: Fundamental gates: AND gate, OR gate, NOT gate (Definition, Symbol, Truth table)
➢ Derived gates: NOR gate, NAND gate, X-OR gate, X-NOR gate (Definition, Symbol, Truth Table), NAND and NOR gates as universal gates. Constructing logic circuits using basic gates and universal gates.
➢ Adder circuits: Half Adder and Full Adder – Definition, Truth table, obtaining simplified expression for sum and carry ,Circuit Diagrams, obtaining full adder from half adders.

# COMPUTER NETWORKS

- **Networks:**
  Definition, Components, Need for Networking, advantages, disadvantages.

- **Types of Networks:**
  LAN, MAN, WAN ( definations)

- **Communication Channel:**
  Physical Channel: Twisted Pair Cable, Co-axial Cable, Optical Fibre Cable (Diagram, description, application).
  Wireless Channel: Microwave, Radio wave, and Satellite Links.

- **Data Switching Techniques:**
  Circuit Switching, Message Switching and Packet Switching.

- **Network Devices and their uses:**
  Modem, Hub, Repeaters, Bridge, Router, Gateway, Switch.

- **Network Topologies:**
  Definition, Types of Topologies with advantages and disadvantages  (Bus, Tree, Star, Ring).

- **Application of Networks:**
  Email, E-commerce, Chat Services, Video Conferencing, Usenet.

- **Protocols:**
  Definition, File Transfer Protocol (FTP), Hyper Text Transfer
  Protocol (HTTP), Transmission Control Protocol/Internet Protocol(TCP/IP), Remote Login(Telnet).

- **Internet Related Terminologies:**
  Internet, Internet Service Providers, Internet Addressing, World Wide Web(WWW), Uniform Resource Locator(URL), Web Server, Web page, Website, Web Browser, Hyper Text Mark-up Language ( HTML), Dynamic Hyper Text Mark-up Language (DHTML), Search Engine, Downloading and Uploading files, Hacking, Cracking,

<h1 style="text-align:center">LIST OF PRACTICALS</h1>

<h2 style="text-align:center">SECTION A - SHORT PROGRAMS</h2>

1. Define a class named **NUMBER** consisting of the following:
   i ) **num**: of type short unsigned integer under private visibility label.
   ii) Define a private member function named "**input**" to accept the data member "num", determine and return 1 if the "num" is palindrome else it return 0.
   iii) Define a public member function named "**display**" which call the member function 'input' and display whether the number is palindrome or not.
   Write a relevant main function to complete the program.

   **NOTE:- A positive integer number is a "Palindrome Number" if its reversal is equal to original number.**

2. Define a class named **FIBO** consisting of the following members:
   i) **N**: of type unsigned short integer under the private visibility label.
   ii) Define a private member function named "**input**" to accept value 'N' from the user.
   iii)Define a public member function named "**display**" which call the member function "input" and display the Fibonacci series up to the 'N' terms.

   Write a relevant main function to complete the program.

   **NOTE:- Fibonacci number is an integer in the infinite series 0,1,1,2,3,5,8………. of which the first two terms are 0 and 1 and each succeeding term is the sum of the previous two terms.**

3. Define a class named **PRIME** consisting of the following members:-
   i) **num**:- of type unsigned short integer under private visibility label
   ii) Define a private member function named **get_no( )** to accept a positive number from the user.
   iii) Define a public member function named **process()** which call the member function get_no(), checks and displays if num is prime or composite number.

   Write a relevant main function to complete the program .

   **NOTE:- A positive integer number is said to be prime if it is only divisible by 1 and itself.**
   **Egs of prime numbers:- 2, 3, 5, 7, 11 ……..**
   **Egs of Composite numbers:- 4, 6, 9, …..**
   **Number 1 is neither prime nor composite.**

4. Define a class named **ARMSTRONG** consisting of the following members:
   i) **num**: of type short unsigned integer under private visibility label.
   ii) Define a public member function "**get_no**" to read a three digit positive integer no from the user and call the member function "process"
   iii) Define a private member function "**process**" which checks and display if num is a Armstrong number or not.

Write a relevant main function to complete the program

**Note:A three digit positive integer number is a "Armstrong Number " if sum of cubes of its digits is equal to the given number itself. Example 153 is a Armstrong number .**
**1\*1\*1+5\*5\*5+3\*3\*3=1+125+27=153**

5. Define a class named **SERIES** consisting of the following Members:
   i) **N:** of type unsigned short integer under private visibility label. (N indicates number of terms)

   ii) **D**: of type float under private visibility label. (D indicates angle in degree)

   iii) Define a public member function "**get_data**" to read values for the data members D and N.

   iv) Define a private member function "**process**" to compute sin(x) by determining the summation of first "N' terms of the following series for display.

   $$\sin(x)= \frac{x}{1!} - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} - \ldots\ldots\ldots\ldots \frac{x^N}{N!} \text{ where x is angle in radians.}$$

   Write a relevant main function to complete the program.


6. Define a class named **SMALLER** consisting of the following members:-
   i)    d1, d2, d3 of type double under private visibility label
   ii)    Define a private member function named " process" which finds the smaller of d1, d2, and d3 and displays the smallest
   iii)    Define a parametrized constructor function which initialises the private data members.

   Write the relevant main function to complete the program.

7. Define a class **BASE** having one private data member num1 and one public data member **num2** both of type float. Define public member functions:
   i)    **input_data**()- to read data value num1.
   ii)    **get_num1**()- to return the value of num1.
   Extend class **BASE** to another class **DERIVED** using public derivation.
   Define for class DERIVED a private data member sum which is to be calculated by adding num1 and num2 and a public member function:
   i)    **get_data**()- to read num2 and to call input_data() for reading value to add to compute sum.
   ii)    **show_data**()-to output num1, num2 and sum.
   Write a main() to create object of type DERIVED and input and output all data.

8. Define an abstract class named "**BASE1**" which has the following members:
   i)      **A** : an integer type variable under private visibility label
   ii)     Define an inline parameterized constructor to initialize data member "A".
   iii)    **Get_A()**: an inline protected member function which returns value of "A".
   Define an abstract class named "**BASE2**" which has the following members:
   i)      B: an integer type variable under private visibility label
   ii)     Define a inline parameterized constructor to initialize data member "B".
   iii)    Get_B: an inline public member function which returns value of "B".
         Define a class named "**DERIVED**" which is derived from "BASE1" and "BASE2"
   under **public** and **protected** mode respectively. It has the following members:
   i)      Z: an integer type variable under private visibility label.
   ii)     show(): an inline private member function which displays value of Z.
   iii)    Define an inline parameterized constructor to initialize data member "Z" with product
           of "A" and "B", it further calls member function show().
           Define an appropriate main function.

9. Define a class named "**TIME**" which has the following members:
   i)    **Hours, Minutes**: of type integer under private visibility label
   ii)   Define a **parameterized constructor** to initialize data members "Hours" and
         "Minutes.
   iii**)** **Sum()**: a public member function which accepts two objects as parameter of type
         "TIME". It calculates summation of two time quantities represented by parameters and
         assigns it to data members of object which has called member function sum().
   iv**)** **Display()**: a public member function to display data members of object of type "TIME".
         Define an appropriate main() function to display addition of two objects of type "TIME".

10) Define a class named **VOLUME** consisting of the following members:
    i) **r,r1,h,v**: of type float under private visibility label.
    ii) Define a **parameterized constructor** to initialize data member "r"
    iii) Define a **parameterized constructor** to initialize data members "r1" & "h".
    iv**)** **Display()**: to display Volume of sphere $(v=4/3\pi r^3)$ and volume of cylinder $(v=\pi r^2 h)$
    Write the relevant main function to complete the program.

# LONG PROGRAMS

1. Define a class named **BINARY** consisting of the following members:
   i) **list**: an array of type short integer of size 30 under private visibility label.
   ii) **N** : of type short unsigned integer (indicates total number of elements to be accepted in array "list" ) under private visibility label.
   iii) Define a **default constructor** to accept data member "N" and to accept the numbers in the array "list". It further calls member function "search".
   iv) Define a private member function named "**search**" which accepts the number to be searched from the user and determines whether it is present in the "list" using **binary search technique**.
   Write a relevant main function to complete the program.

2. Define a class named **BUBBLE** consisting of the following members:
   i) **list**: an array of type short integer of size 30 under private visibility label.
   ii) **N** : of type short unsigned integer(indicates total number of elements to be accepted in array "list")under private visibility label.
   iii) Define a **default constructor** to accept data member "N" and to accept the numbers in the array "list" .It further calls member function "sort" followed by member function "show".
   iv) Define a private member function named "**sort**" which performs sorting of numbers in array "list" using **bubble sort technique**(sort in **ASCENDING** order).
   v) Define a private member function named "**show**" which displays the content of array.
   Write a relevant main function to complete the Program.

3. Define a class named **SELECT** consisting of the following members:
   i) **list:** an array of type short integer of size 30 under private.
   ii) **N** :of type short unsigned integer.(indicates total number of elements to be accepted in array "list" ) under private .
   iii) Define a **default constructor** to accept data member "N" and to accept the numbers in the array "list". It further calls member function "sort" followed by member function "show".
   iv) Define a private member function named "**sort**" which performs sorting of numbers in the array "list" using **selection sort technique** (sort in **ASCENDING** order).
   v) Define a private member function named "**show**" which displays the content of array.
   Write a relevant main function to complete the program.

4. Define a class named **INSERT** consisting of the following members:
   i) **list**: an array of type short integer of size 30 under private visibility label.
   ii) **N** : of type short unsigned integer(indicates total number of elements to be accepted in array "list" ) under private visibility label.
   iii) Define a **default constructo**r to accept data member "N" and to accept the numbers in the array "list". It further calls member function "sort" followed by member function "show".
   iv) Define a private member function named "**sort**" which performs sorting of numbers in the array "list" using **insertion sort** technique (sort in **ASCENDING** order).

v) Define a private member function named "**show**" which displays the content of array. Write a relevant main function to complete the program.

5. Define a class named **MERGE** consisting of the following members:
   i) **A,B,C**: 1-D arrays of type short integer of size 50 each under private visibility label.
   ii) **M,N** : of type short unsigned integer (indicates total number of elements in A and B respectively ) under private visibility label.
   iii) Define a **default constructor** to accept data members "M" and "N" and to accept the numbers in the arrays A and B (both in **ascending** order). It further calls member function "process".
   iv) Define a private member function named "**process**" which performs **merging** of all elements in A and B to obtain array C in **ASCENDING** order for display.
   Write a relevant main function to complete the program.

6. Define a class named **MATRIX** consisting of the following members:
   i) **M.N,Q**:of type short unsigned integer under private visibility label.

   ii) **A,B,C**: 2-D arrays of size 10x10 each under private visibility label.

   iii) Define a **default constructor** to accept data members M,N and Q (Where MxN is size of A and NxQ is size of B). It also accepts 2-D array A and B. It further calls member function "product".
   iv) Define a private member function named "**product**" which computes product of two matrix's A and B and stores in C. It further displays matrix C in tabular form.

   Write a relevant main function to complete the program.

7. Define a class named **BILL** with the following members:
   i) **item_code**: of type unsigned short integer under private visibility label.

   ii) **item _name**: a character array of size 30 under private visibility label.

   iii**) unit_price,total** : of type float under private visibility label.

   iv) **quantity**: of type unsigned short integer under private visibility label.

   v) Define a member function named "**get data**" to accept data members item_code, item_name,unit_price and quantity. It computes total as quantity * unit_price.
   vi) Define a member function named "**put data**" to display data members item_code, item_name,unit _price,quantity and total.
   Write a menu driven main function to
   a) **Create** a binary file named "market.data" containing objects of type BILL.
   b) **Display** all the data members of the objects read from file "market.data" in tabular form.
   SAMPLE OUTPUT

| ITEM CODE | ITEM NAME | UNIT-PRICE | QUANTITY | TOTAL |
|---|---|---|---|---|
| 1000 | PEN | 10.00 | 5 | 50.00 |
| 2000 | PENCIL | 5.00 | 3 | 15.00 |

8.  Write a menu driven program to implement a **singly linked list** in which each node consists of the following data fields:

i) **M.N,Q**: of type short unsigned integer under private visibility label.

ii) **roll**- of type unsigned short integer.

iii)**name**- an array of maximum 30 characters.

iv)**percent**- of type float

And perform the following operations:

a) **Creation** of a linear linked list containing "n" nodes.

b) **Display** the linear linked list in the following format:

| ROLL | NAME | PERCENT |
|------|------|---------|
| 1000 | SACHIN | 89.9 |
| 2000 | KARUN | 75.52 |
| 3000 | KIERA | 65.18 |

9.  Write a menu driven program to implement a **stack** using singly linked list in which each node consists of a single data field of type **character** and performs the following operations:

i)      **push** a node onto the stack

ii)     **pop** the top node from the stack

Displaying data field of all the nodes in the stack horizontally.

10. Write a menu driven program to implement a **queue** using singly linked list in which each node consists of a single data field of type character and performs the following operations:

i)      **Appending** a node into the queue.

ii)     **Deleting** the first node in the queue.

Displaying data field of all the nodes in the queue horizontally.

# DESIGN OF THE QUESTION PAPER (FIRST MID-TERM 2024-25)
## CLASS : XII (General Stream)

Time : 1. Hrs.        Subject : Computer Science(H-4705)        Max. Marks :20

The weightage or the distribution of marks over different dimensions of the question paper shall be as follows:

1. Weightage to Learning Outcomes

| Sr.No. | Learning Outcomes | Marks | Percentage of marks |
|---|---|---|---|
| 1. | Knowledge | 04 | 20% |
| 2. | Understanding | 10 | 50% |
| 3. | Application | 06 | 30% |
| 4. | Skill | --- | --- |
| | Total | 20 | 100% |
| | | | |

2. Weightage to Content / Subject Units

| Sr.No. | Units | Marks |
|---|---|---|
| 1. | Programming in C++( Sub units-01--09) | 10 |
| 2. | Boolean Algebra | 10 |
| | Total | 20 |

3.Weightage to Forms of question

| Sr. No. | Forms of Question | Marks for each Question | Number of Questions | Total Marks |
|---|---|---|---|---|
| 1. | Long Answer Type (LA) | 04 | --- | --- |
| 2. | Short Answer Type (SA-I) | 02 | 03 | 06 |
| 3. | Short Answer Type (SA-II) | 03 | 03 | 09 |
| 4. | Very Short Answer Type (VSA) **( 2 MCQ's, one from each unit)** | 01 | 05 | 05 |
| | Total | | 11 | 20 |

The expected time for different type of question would be as follows:

| Sr. No | Forms of Question | Approx. Time for each Question in | Number of Questions | Approx. Time for each Question in |
|---|---|---|---|---|

| | | mins (t) | | mins (n x t) |
|---|---|---|---|---|
| 1. | Long Answer Type (LA) | --- | --- | --- |
| 2. | Short Answer Type (SA-I) | 06 | 05 | 30 |
| 3. | Short Answer Type (SA-II) | 10 | 02 | 20 |
| 4. | Very Short Answer Type (VSA) | 02.5 | 04 | 10 |
| | Total | | 11 | 60 |

As the total time is calculated on the basis of the number of questions required to be answered and the length of their anticipated answers, it would therefore be advisable for the candidates to budget their time properly by cutting out the superfluous words and be within the expected time limits.

4. **Scheme of Options**

There will be no overall choice. However, there is an internal choice for question numbers 11(Unit 1).

5. **Weightage to difficulty level questions:**

| Sr.No. | Estimated difficulty level of questions | Percentage |
|---|---|---|
| 1. | Easy | 20% |
| 2. | Average | 60% |
| 3. | Difficulty | 20% |

# DESIGN OF THE QUESTION PAPER (FIRST TERM 2024-25)
## CLASS : XII (General Stream)

Time : 150 minutes          Subject : Computer Science (H-4705)          Max. Marks :60


The weightage or the distribution of marks over different dimensions of the question paper shall be as follows:

1. Weightage to Learning Outcomes

| Sr.No. | Learning Outcomes | Marks | Percentage of marks |
|--------|-------------------|-------|---------------------|
| 1. | Knowledge | 12 | 20% |
| 2. | Understanding | 30 | 50% |
| 3. | Application | 18 | 30% |
| 4. | Skill | --- | --- |
| | Total | 60 | 100% |
| | | | |

2. Weightage to Content / Subject Units

| Sr.No. | Units | Marks |
|--------|-------|-------|
| 1. | Programming in C++ | 25 |
| 2. | Data structures using C++( Sr. No. 1 & 2) | 15 |
| 3. | File Handling using C++ | 10 |
| 4. | Boolean Algebra | 10 |
| | Total | 60 |

3.Weightage to Forms of question

| Sr. No. | Forms of Question | Marks for each Question | Number of Questions | Total Marks |
|---------|-------------------|-------------------------|---------------------|-------------|
| 1. | Long Answer Type (LA) | 04 | 02 | 08 |
| 2. | Short Answer Type (SA-I) | 02 | 13 | 26 |
| 3. | Short Answer Type (SA-II) | 03 | 04 | 12 |
| 4. | Very Short Answer Type (VSA) **( 7 MCQ's)** | 01 | 14 | 14 |
| | Total | | 33 | 60 |

The expected time for different type of question would be as follows:

| Sr. No | Forms of Question | Approx. Time for each Question in mins (t) | Number of Questions | Approx. Time for each Question in mins (n x t) |
|---|---|---|---|---|
| 1. | Long Answer Type (LA) | 12.5 | 02 | 25 |
| 2. | Short Answer Type (SA-I) | 05 | 13 | 65 |
| 3. | Short Answer Type (SA-II) | 08 | 04 | 32 |
| 4. | Very Short Answer Type (VSA) | 02 | 14 | 28 |
| | Total | | 33 | 150 |

As the total time is calculated on the basis of the number of questions required to be answered and the length of their anticipated answers, it would therefore be advisable for the candidates to budget their time properly by cutting out the superfluous words and be within the expected time limits.

4. **Scheme of Options**
There will be no overall choice. However, there is an internal choice for question numbers 31(3 marks), 32(4 marks), 33(4 marks).

6. **Weightage to difficulty level questions:**

| Sr.No. | Estimated difficulty level of questions | Percentage |
|---|---|---|
| 1. | Easy | 20% |
| 2. | Average | 60% |
| 3. | Difficulty | 20% |

# DESIGN OF THE QUESTION PAPER
## (Final Exam 2024-25 ONWARDS)
### CLASS : XII (General Stream)

Time : 3 Hrs.              Subject : Computer Science  (H-4705)     Max. Marks :70

The weightage or the distribution of marks over different dimensions of the question paper shall be as follows:

1. Weightage to Learning Outcomes

| Sr.No. | Learning Outcomes | Marks | Percentage of marks |
|---|---|---|---|
| 1. | Knowledge | 21 | 30% |
| 2. | Understanding | 35 | 50% |
| 3. | Application | 14 | 20% |
| 4. | Skill | --- | --- |
| | Total | 70 | 100% |

2. Weightage to Content / Subject Units

| Sr.No. | Units | Marks |
|---|---|---|
| 1. | Programming with C++ | 20 |
| 2. | Data Structures | 20 |
| 3. | File Handling in C++ | 10 |
| 4 | Boolean Algebra | 10 |
| 5 | Computer Networks | 10 |
| | Total | 70 |

3. Weightage to Forms of question

| Sr.No. | Forms of Question | Marks for each Question | Number of Questions | Total Marks |
|---|---|---|---|---|
| 1 | Very Short Answer Type (VSA) (MCQ's : 7) | 01 | 14 | 14 |
| 2 | Short Answer Type (SA-I) | 02 | 10 | 20 |
| 3 | Short Answer Type (SA-II) | 03 | 8 | 24 |
| 4 | Long Answer Type (LA) | 04 | 3 | 12 |
| | Total | | 35 | 70 |

The expected time for different type of question would be as follows:

| Sr. No | Forms of Question | Approx. Time for each Question in mins (t) | Number of Questions | Approx. Time for each Question in mins (n x t) |
|--------|-------------------|---------|-----------|---------|
| 1 | Very Short Answer Type (VSA) | 2 | 14 | 28 |
| 2 | Short Answer Type (SA-I) | 4 | 10 | 40 |
| 3 | Short Answer Type (SA-II) | 6 | 8 | 48 |
| 4 | Long Answer Type (LA) | 11 | 3 | 34 |
| | Total | | 35 | 150 |

As the total time is calculated on the basis of the number of questions required to be answered and the length of their anticipated answers, it would therefore be advisable for the candidates to budget their time properly by cutting out the superfluous words and be within the expected time limits.

## 4. Scheme of Options

There will be no overall choice. However, there is an internal choice for question numbers 25,29,34 and 35.

## 5. Weightage to difficulty level questions:

| Sr.No. | Estimated difficulty level of questions | Percentage |
|--------|------------------------------------------|------------|
| 1. | Easy | 20% |
| 2. | Average | 60% |
| 3. | Difficulty | 20% |

# MODEL PAPER FOR BOARD EXAM
## SUBJECT : COMPUTER-SCIENCE
## SUBJECT CODE : 4705

**DURATION : 3 HOURS**　　　　　　　　　　　　　　　　**MAX. MARKS : 70**
**TOTAL NO. OF QUESTIONS : 35**　　　　　　　　　**NO. OF PRINTED PAGES : 10**

**Instructions:**

　　i)　　**All questions are compulsory, however there is an internal choice for question number 31, 32, 34, 35**

　　ii)　　**All questions should be attempted only once.**

　　iii)　　**Programs should be written in C++ only.**

　　iv)　　**State your assumptions clearly**

　　　　Section A : Consists of 14 questions of 1 mark each

　　　　Section B : Consists of 10 questions of 2 mark each

　　　　Section C : Consists of 08 questions of 3 mark each

　　　　Section D : Consists of 03 questions of 4 mark each

==========================================================================

## SECTION A

| 1 | Write the **CORRECT** alternative from those given below: <br> If a function is declared as <br> 　　void func(int a=0, float b=9.8, int c=76); <br> Then which of the following is an **INCORRECT** method of calling the function? | 1 |
|---|---|---|

| • func(); | • func(34); |
|---|---|
| • func(6.7); | • func(12,8.6,34); |

| 2 | Write the **CORRECT** alternative from those given below: <br> Suppose a 2-dimensional array of 15 integers X[3][5] is stored in row major order. How many integers will be stored in between A[1][0] & A[2][4] (both these not included in the count)? | 1 |
|---|---|---|

| • 10 | • 8 |
|---|---|
| • 15 | • 9 |

| 3 | The postfix form of the expression (A+B)*(C*D-E)*F/G is _____ | 1 |
|---|---|---|

| • AB+CD*E-FG/** | • AB+CD*E-F**G/ |
|---|---|
| • AB+CDE*-*F*G/ | • AB+CD*E-*F*G/ |

| 4 | Write the **CORRECT** alternative from those given below: <br> Which stream class is used only to write to the files? | 1 |
|---|---|---|

| • Ofstream | • ifstream |
|---|---|
| • fstream | • iostream |

| | | |
|---|---|---|
| | | |
| 5 | Write the **CORRECT** alternative from those given below:<br>A+AB+ABC+ABCD+ABCDE+…….. = _____ | 1 |
| | • 1            • *A*<br>   • A+AB        • *A.B* | |
| 6 | Write the **CORRECT** alternative from those given below:<br>Which of the following is BUS topology?<br><br>Fig (a)    Fig (b)    Fig (c)    Fig (d) | 1 |
| | • Fig a        • Fig c<br>• Fig b        • Fig d | |
| 7 | Write the **CORRECT** alternative from those given below:<br>Which of the following devices forwards packets between networks by processing the routing information included in the packet? | 1 |
| | • Firewall        • bridge<br>• router          • hub | |
| 8 | What is an inline function? | 1 |
| 9 | Define the term data encapsulation. | 1 |
| 10 | Why is a queue known as a FIFO data structure? | 1 |
| 11 | How is the put pointer different from get pointer w.r.t. files? | 1 |
| 12 | Define an inverter. | 1 |
| 13 | What is protocol? | 1 |
| 14 | Write the function of a repeater in a computer network. | 1 |
| | | |

## SECTION B

| 15 | Consider the following class declaration and answer the questions given below it: | 2 |
|---|---|---|

```cpp
class testmeout
{
    int rollno;
    public:
    testmeout()                                    //Function 1
    { rollno=10;
      cout<<rollno<<"  is appearing for exam";
    }
    testmeout(testmeout &t) { //missing code }      //Function 2
    ~testmeout()                                   //Function 3
    {cout<<endl<<rollno<<" is leaving the exam hall";}
    void mywork()
    { cout<<endl<<rollno<<" is writing the exam";}
};
```

i)   What is function 2 called in context to OOPS.
ii)  Write the missing code for function 2.
iii) What will be the output of the above code in case if the main function contains the following statements?

```cpp
int main()
{
    testmeout ob1;
    ob1.mywork();
}
```

| 16 | Determine the output of the following C++ program: | 2 |
|---|---|---|

```cpp
#include<iostream>
using namespace std;
void fun(int &x, int &y)
{
        if(x>y)
                x++;
        else if(y>x)
                y-=2;
```

```
        else
        {  x--; y++;}
    }
    int main()
    {
        int a=10,b=5, c=7;
        fun(a,b);
        cout<<endl<<a<<" "<<b;
        fun(c,a);
        cout<<endl<<c<<" "<<a;
        fun(b,c);
        cout<<endl<<c<<" "<<b;
        fun(a,c);
        cout<<endl<<a<<" "<<c;
    }
```

| 17 | State **TWO** points of difference between linear search & binary search. | 2 |
|----|--------------------------------------------------------------------------|---|
| 18 | Sort the following numbers in ascending order using selection sort and show the contents of the array at the end of each pass.<br>55, 99,21,12,45 | 2 |
| 19 | A binary file student.dat exists, containing objects of class STUDENT type. Assuming that the file has just been opened using object fil of ifstream class type, answer the following questions:<br>　i)　　Write a single statement in C++ to place the file pointer to the 9th record(student) from the beginning.<br>　ii)　　Write a single statement in C++ to place the file pointer 2 records (student) backward from the current position | 2 |
| 20 | Write a function in C++ to count and return the number of words starting with a capital letter present in the text file data.txt. | 2 |
| 21 | Simplify using laws of Boolean Algebra<br>$$AB + (\overline{AC}) + A\bar{B}C(AB + C)$$ | 2 |
| 22 | State any two points of differences between LAN & WAN. | 2 |
| 23 | Write a short note on packet switching. | 2 |

| 24 | Identify the following: | 2 |
|---|---|---|
| | i)  A protocol used to transfer hypertext documents on the internet. | |
| | ii)  A transmission medium used to transfer data across two or more continents at very high speed. | |

<div align="center"><strong>SECTION C</strong></div>

| 25 | Obtain the simplified Boolean expression in SOP form for the following Boolean function<br><br>F(X,Y,Z) = Π(3,6)<br><br>Also Draw the logic circuit diagram for the simplified expression using NAND gates only. | 3 |
|---|---|---|
| 26 | Define a class FIGURE in C++ with following specifications :<br> Private members:<br>  - sides  //No. of sides<br>  - Shape // Name of the Shape<br>  - AssignShape() - function To assign value of Shape based upon Sides as follows: | 3 |

| Sides | Shape |
|---|---|
| Less than 3 | Open |
| Equal to 3 | Triangle |
| Equal to 4 | Quadrilateral |
| More than 5 | Polygon |

Public members:
 - GetFigure() - function to allow user to enter value of  Sides. Also, this method should call function Assign() to assign value of Shape
- ShowFigure() – function to display Sides and Shape

| 27 | Answer the questions (i) to (iv) based on the following : | 3 |
|---|---|---|

```
class Teacher
{
    int TCode;
    protected:
    char Name[20];
     public:
      Teacher();
      void Enter();
      void Show();
```

```
};
class Course
{
    int ID;
    protected:
     Char Title[30];
    public:
    Course();
    void Initiate();
    void Display();
};
class Schedule : public Course, private Teacher
{
    int DD,MM,YYYY;
    public:
    Schedule();
    void Start();
    void View();
};
void main()
{
    Schedule S;
}
```

   (i)     Which type of Inheritance out of the following is illustrated in the above example ?

   (ii)    Write the names of all the members, which are directly accessible by the member function View() of class Schedule.

   (iii)   Write the names of all the members, which are directly accessible by the object S of class Schedule declared in the main() function.

   (iv)   What will be the order of execution of the constructors, when the object S of class Schedule is declared inside the main() function ?

| 28 | Evaluate the following postfix expression showing the stack content after every step.<br><br> 3,12,/,9,2,*,+,30,-,14,7,/,+ | 3 |
| 29 | Obtain simplified SOP Boolean expression for the following expression  using K- | 3 |

| | map and draw the logic circuit diagram for the simplified expression using basic gates only<br><br>F(X,Y,Z,W)= Σ (0,1,2,3,4,5,8,10,11,14) | |
|---|---|---|
| 30 | Write a definition for a function matrixsum(int A[ ][10], int r) in C++, which returns the sum of the positive even numbers present in the lower triangular matrix A of order rxr, as shown in the example given below:<br>For example, if the content of the array A of order 4x4 is as follows :<br>12 41 -32 73<br>20 -19 18 17<br>15 -14 13 12<br>10 -91 -18 17<br>The function should return 12+ 20+18+12+10=70 | 3 |
| 31 | Write a complete procedural C++ program to display the following pattern for the n lines, if n=5 then the pattern should be<br>*****<br>\*　　\*<br>\*　　\*<br>\*　　\*<br>***** | 3 |
| **OR** | | |
| 31 | Write a complete procedural C++ program to display the following pattern for n lines, if n=5 then the pattern should be<br>ABCDE<br>　B<br>　　C<br>　　　D<br>ABCDE | 3 |
| 32 | Write a function SPLIT(int A[ ], int Size) in C++ to copy all the prime numbers from the array A to array PRIME. The array PRIME should be local to the function.<br>For example:<br>If the contents of array A are<br>11, 16, 19, 17, 26, 15, 9, 7, 13, 1<br>The contents of array PRIME should be<br>11,19,17,7, 13 | 3 |
| | **OR** | |

| 32 | For example: Write a user defined function in C++ to accept an array A of size n and type integer, the function should return the number of duplicate numbers present in the array.<br><br>If the contents of array A are<br><br>11,67,34,11,33,22,33,88,67,11<br><br>Then the function should return 3 | 3 |
|---|---|---|
| 33 | Declare an abstract class named **"PRODUCT"** which has following members.<br>pcode- product code of type integer under private visibility.<br>Define a constructor to initialize the data member pcode.<br>Show1() - A protected function to display data member pcode.<br>Declare a class named **"WHOLESALER"** consisting of the following members.<br>wcode-wholesaler code of type integer and mname – manager name of type 30 characters both under private visibility mode.<br>Define a constructor to initialize the data members wcode & mname.<br>Show2() - A protected function to display data member wcode & mname.<br>Derive a class SHOWROOM above two classes using public derivation. Class SHOWROOM also contains the following members<br>city – name of the city of the showroom of type 20 characters & under private visibility.<br>Define a constructor to initialize the data members city.<br>Show3()- A public function to display city name .<br>Write a main program to accept input values and pass it as arguments to an appropriate constructor. Also display all the data members of both the classes | 4 |
| 34 | Write a function in C++ called delstud(int r)  to remove a student with roll number r from the binary file "STUDENT.dat", assuming the binary file is containing the objects of the following class :<br>class STUDENT<br>{<br>   int Rno;<br>   char Sname[20];<br>   public:<br>   void Enter()  { cin>>Rno;gets(Sname); }<br>   void show() { cout << Rno<<sname<<endl;}<br>   int getroll() {return Rno;}<br>}; | 4 |
| **OR** | | |

| 34 | Write a function in C++ to search and display details of BUS, whose destination is "Cochin" from binary file "Bus.Dat". Assume that the binary file is containing the objects of the following class:<br><br>class BUS<br>{<br>  int Bno; // Bus Number<br>  char From[20]; // Bus Starting Point<br>  char To[20]; // Bus Destination<br>  public:<br>  char * StartFrom ( ); { return From; }<br>  char * EndTo( ); { return To; }<br>  void input()<br>  { cin>>Bno>>; gets(From); gets(To); }<br>  void show( ) { cout<<Bno<< ":"<<< ":" <<To<<endl;}<br>}; | 4 |
| --- | --- | --- |
| 35 | Consider the following class declaration:<br><br>class LIST<br>{<br>   struct node<br>  {    int data;<br>      node *next;<br>  }*start;<br>  public:<br>   LIST() { start=NULL; }<br>   void insert();<br>};<br><br>Write a function definition for insert() to insert a node at position P$(1 \leq P \leq n+1)$ in the linked list. Assume that list contains n nodes. | 4 |
| | **OR** | |
| 35 | Consider the following class declaration:<br>class LIST<br>{ | 4 |

```
    struct node
    {
        float data;
        node *next;
     }*start;
    public:
     LIST() { start=NULL; }
     void  dellist();
};
```

Write a function definition for dellist() to delete a node at position P($1 \leq P \leq n$) from the linked list. Assume that list contains n nodes.