

Maharashtra Board Class 12 Computer Science (P1-D-9-V-264) 2024 Question Paper with Solutions

Time Allowed :3 Hours	Maximum Marks :50	Total Questions :5
-----------------------	-------------------	--------------------

General Instructions

Read the following instructions very carefully and strictly follow them:

1. All questions are compulsory.
2. Figures to the right indicate full marks.
3. Use of any type of calculator not allowed.
4. Draw a neat diagram whenever necessary.

1. Select and write the correct answer for the following multiple-choice questions:

(a) _____ tag is used for scrolling the text on a WebPage.

- (1) <Head>
- (2)
- (3) <MARQUEE>
- (4) <SUP>

Correct Answer: (3) <MARQUEE>

Solution:

The <MARQUEE> tag is used in HTML to create a scrolling effect for text or images on a webpage. Other options:

- <Head>: Defines metadata.
- : Creates an unordered list.
- <SUP>: Displays superscript text.

💡 Quick Tip

The <MARQUEE> tag is now deprecated in HTML5. Use CSS animations instead.

(b) _____ is a free operating system.

- (1) Unix
- (2) Windows
- (3) LINUX
- (4) DOS

Correct Answer: (3) LINUX

Solution:

Linux is an open-source and free operating system that is widely used in servers, desktops, and embedded systems.

Other options:

- Unix: Proprietary system, though Linux is based on it.
- Windows: Paid OS by Microsoft.
- DOS: Older OS, not free.

 Quick Tip

Linux is free and open-source, with distributions like Ubuntu, Fedora, and Debian.

(c) _____ is a mechanism to achieve run-time polymorphism.

- (1) Operator Overloading
- (2) Virtual Function
- (3) Function Overloading
- (4) Constructor

Correct Answer: (2) Virtual Function

Solution:

Virtual functions allow run-time polymorphism in C++ by enabling function overriding in derived classes.

Other options: - Operator Overloading & Function Overloading: Compile-time polymorphism.
- Constructor: Used for object initialization, not polymorphism.

 Quick Tip

Virtual functions allow dynamic binding in C++ and enable polymorphism.

(d) Finding the location of a given element is called as _____.

- (1) Searching
- (2) Sorting
- (3) Traversing
- (4) Inserting

Correct Answer: (1) Searching

Solution:

Searching is the process of finding the location of a specific element in a data structure such as an array, linked list, or tree.

Other options:

- Sorting: Arranges elements in a specific order.
- Traversing: Accesses all elements sequentially.
- Inserting: Adds a new element to the data structure.

 Quick Tip

Searching algorithms include Linear Search and Binary Search, used to locate elements efficiently.

1(B). Answer any two of the following :

(a) Explain the following components of GUI:

(i) Menu Bar

Solution:

A Menu Bar is a horizontal strip at the top of a window or application that contains menus with various options for user interaction.

Features:

- Provides access to different commands and settings.
- Typically contains File, Edit, View, Help, etc.
- Helps users navigate and interact with software easily.

 Quick Tip

Menu Bars enhance usability by organizing functions into easily accessible categories.

(ii) Scroll Bar

Solution:

A Scroll Bar allows users to navigate through content that extends beyond the visible area of a window.

Features:

- Can be vertical (for up-down movement) or horizontal (for left-right movement).
- Contains a slider to move through content efficiently.
- Used in applications like browsers, documents, and file explorers.

 Quick Tip

Scroll Bars improve navigation in applications by enabling movement within large content areas.

(iii) Title Bar

Solution:

A Title Bar is the topmost bar of a window that displays the name of the application or document.

Features:

- Contains the window control buttons (Minimize, Maximize, Close).
- Helps users identify which window they are currently working on.
- Typically found in all graphical operating systems like Windows, macOS, and Linux.

 Quick Tip

Title Bars help users recognize open applications and manage window controls.

1(B).

(b) Write three distinct differences between Array and Linked List.

Solution:

Arrays and linked lists are two fundamental data structures used for storing elements. Below are the key differences:

Feature	Array	Linked List
Storage	Contiguous memory allocation	Dynamic memory allocation
Insertion/Deletion	Costly, requires shifting elements	Efficient, modifies pointers
Access Time	Fast (direct access via index)	Slow (sequential traversal)

 Quick Tip

Arrays are preferred for fast access due to direct indexing, while linked lists are useful for dynamic memory allocation and frequent insertions/deletions.

(c) Write any three advantages and disadvantages of HTML.

Solution:

HTML (HyperText Markup Language) is the fundamental language for structuring web pages. Below are its advantages and disadvantages:

Advantages of HTML:

1. **Easy to Learn** – Simple syntax and widely supported.
2. **Platform Independent** – Works across different devices and browsers.
3. **SEO Friendly** – Helps in improving search engine rankings.

Disadvantages of HTML:

1. **Static Nature** – Cannot handle dynamic data without JavaScript.
2. **Limited Functionality** – Requires CSS and JavaScript for advanced features.
3. **Longer Code for Complex Pages** – Writing large websites in plain HTML can be inefficient.

 Quick Tip

HTML is the foundation of web pages, but for interactive elements, CSS and JavaScript are required.

2(A). Answer any two of the following:

(a) Define Data Structure. Explain any four operations of Data Structure.

Solution:

Definition: A Data Structure is a way of organizing and storing data in a computer so that it can be used efficiently. Examples include arrays, linked lists, stacks, queues, trees, and graphs.

Four Basic Operations of Data Structures:

1. Insertion – Adding an element to a data structure (e.g., inserting an element in an array).
2. Deletion – Removing an element from a data structure (e.g., deleting a node from a linked list).
3. Searching – Finding an element in a data structure (e.g., searching for a key in a binary search tree).
4. Sorting – Arranging data in a specific order (e.g., ascending or descending order using sorting algorithms like Merge Sort).

💡 Quick Tip

Data Structures help in efficient data management, enabling quick access, modification, and storage.

(b) Write any three distinct differences between Procedure-Oriented and Object-Oriented Programming.

Solution:

Procedure-Oriented Programming (POP) and Object-Oriented Programming (OOP) are two different paradigms of programming.

Feature	Procedure-Oriented Programming (POP)	Object-Oriented Programming (OOP)
Approach	Focuses on functions and procedures	Focuses on objects and data
Data Security	Data is global and accessible by functions	Data is encapsulated within objects
Code Reusability	Code reusability is limited	Supports inheritance for code reuse
Example	C, Pascal	C++, Java, Python

💡 Quick Tip

OOP enhances modularity, code reusability, and security, while POP follows a step-by-step procedural approach.

(c) Write an algorithm to insert an element into a linear array.

Solution:

The following algorithm inserts an element into a linear array at a given position.

Algorithm: Insert an Element into a Linear Array

1. Start.
2. Read the array A, its size N, the element X to insert, and the position P.
3. If $P < 1$ or $P > N+1$, print "Invalid Position" and exit.
4. Shift all elements from position P onwards one step to the right.
5. Insert X at position P.
6. Increase the size N by 1.
7. Stop.

Example Implementation (C Code): ““c

include <stdio.h>

```
void insertElement(int A[], int *N, int X, int P) {  
    if (P < 1 — P > *N + 1)  
        printf("Invalid Position  
n");  
    return;
```

```
    for (int i = *N; i >= P; i--)  
        A[i] = A[i - 1];  
    A[P - 1] = X;  
    (*N)++;  
}
```

```
int main() {  
    int A[10] = {1, 2, 3, 5, 6}, N = 5, X = 4, P = 4;  
    insertElement(A, &N, X, P);  
    for (int i = 0; i < N; i++)  
        printf("%d ", A[i]);  
    return 0; }
```

💡 Quick Tip

Inserting an element in an array requires shifting elements, which makes it an $O(N)$ operation.

2(B). Answer any one of the following :

2(B)(a). Define the following terms related to the Operating System:

(i) External Priority

Solution:

External Priority refers to the priority assigned to a process based on external factors, such as:

- User-defined priority levels.
- System policies (e.g., priority for critical applications).
- Job scheduling algorithms in multi-user systems.

 Quick Tip

External priority is often assigned by the operating system scheduler based on system policies or user preferences.

(ii) Internal Priority

Solution:

Internal Priority is the priority assigned dynamically by the operating system based on:

- CPU usage (e.g., processes using more CPU may have reduced priority).
- Waiting time (e.g., processes waiting longer may get higher priority).
- Process behavior (e.g., I/O-bound vs CPU-bound processes).

 Quick Tip

Internal priority helps in fair CPU allocation by dynamically adjusting process priority based on system conditions.

(iii) Purchase Priority

Solution:

Purchase Priority is not a standard OS term. It may refer to:

- Resource allocation priority in cloud computing environments.
- Licensing-based priority where premium users get priority access.

 Quick Tip

Operating systems usually focus on process priority scheduling, whereas purchase priority is more relevant in cloud-based services.

(iv) Time Slice

Solution:

A Time Slice is the fixed time allocated to a process in a time-sharing operating system. It determines how long a process can execute before the CPU switches to another process.

Characteristics:

- Used in Round Robin Scheduling.
- Helps in fair CPU time allocation.
- Reduces CPU starvation by ensuring all processes get execution time.

 Quick Tip

A smaller time slice results in faster process switching, improving responsiveness, but may increase context switching overhead.

2(B).

(b) Write declaration statements in C++ for:

(i) Pointer to float type

Solution:

A pointer to a float type in C++ is declared as:

```
float *ptr;
```

Here, `ptr` is a pointer that stores the address of a float variable.

 Quick Tip

In C++, pointers store memory addresses. Use `*` to declare a pointer and `&` to get an address.

(ii) An array of 10 real numbers

Solution:

An array of 10 real (floating-point) numbers in C++ is declared as:

```
float arr[10];
```

This declaration creates an array `arr` of size 10 that can store floating-point values.

 Quick Tip

Arrays in C++ are zero-indexed, meaning the first element is accessed as `arr[0]`.

(iii) An array of 80 characters

Solution:

An array of 80 characters (string storage) in C++ is declared as:

```
char str[80];
```

This array `str` can store a string of up to 79 characters (leaving space for `\0` termination).

 Quick Tip

Character arrays are used for string handling in C++. Use `cin.getline(str, 80);` for input.

(iv) **An object of class Student**

Solution:

An object of a class `Student` in C++ is declared as:

```
class Student {  
    public:  
        string name;  
        int age;  
};  
  
Student obj;
```

Here, `obj` is an instance of the `Student` class.

 Quick Tip

In C++, objects are instances of a class, containing attributes (variables) and methods (functions).

3(A). Answer any two of the following:

(a). Explain the following tags with examples:

**(i)
**

Solution:

The '`<BR>`' (Break) tag is an HTML element used to insert a line break in text content.

Example:

```
This is line 1. <BR>  
This is line 2.
```

Output: This is line 1.

This is line 2.

 Quick Tip

The '`<BR>`' tag is a self-closing tag that does not need a closing '`</BR>`'.

(ii) <PRE> Tag

Solution:

The '`<PRE>`' (Preformatted) tag preserves spaces, line breaks, and indentation in text exactly as written.

Example:

```
<PRE>
  This is
    preformatted
    text.
</PRE>
```

Output: This is preformatted text.

💡 Quick Tip

The '`<PRE>`' tag is useful for displaying code snippets or formatted text as it appears in the HTML source.

(iii) `<U>` Tag

Solution:

The '`<U>`' (Underline) tag is used to underline text in an HTML document.

Example:

```
<U>This is underlined text.</U>
```

Output: `<u>This is underlined text.</u>`

💡 Quick Tip

The '`<U>`' tag was deprecated in HTML5. Instead, use CSS: `text-decoration: underline;`

3.

(b) What is the function of Information Management? List any four system calls in it.

Solution:

Function of Information Management: Information management in an operating system involves handling and maintaining system-related information, including files, memory, processes, and security policies.

Major Functions:

1. File Management – Creating, deleting, and accessing files.
2. Process Management – Scheduling, creating, and terminating processes.
3. Memory Management – Allocating and deallocating memory.
4. Security & Protection – Managing permissions and preventing unauthorized access.

Four System Calls in Information Management:

1. `open()` – Opens a file for reading or writing.
2. `read()` – Reads data from a file.
3. `write()` – Writes data to a file.
4. `close()` – Closes an open file.

💡 Quick Tip

System calls act as an interface between user programs and the operating system to execute critical operations securely.

3. (c) What is a Constructor? Write any four rules for a constructor function.

Solution:

Definition of Constructor: A constructor is a special member function in C++ that automatically initializes an object when it is created. It has the same name as the class and does not return any value.

Four Rules for Constructor Function:

1. Same Name as Class – The constructor must have the same name as the class.
2. No Return Type – Constructors do not return any value, not even 'void'.
3. Automatically Invoked – A constructor is called automatically when an object is created.
4. Can be Overloaded – Multiple constructors with different parameters can exist in a class.

Example:

```
class Student {
public:
    Student() {
        cout << "Constructor Called";
    }
};

int main() {
    Student obj;
    return 0;
}
```

Output: Constructor Called

💡 Quick Tip

Constructors simplify object initialization by automatically assigning values without calling a function explicitly.

3(B). Answer any one of the following :

(a) What is a Binary Tree? Draw a Binary Tree for the following expression:

$$Y = \frac{(A + B)}{(C - D)^2}$$

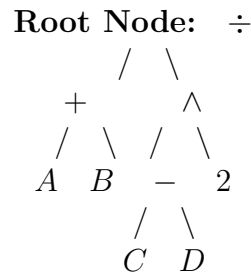
Solution:

Definition of Binary Tree: A Binary Tree is a hierarchical data structure where each node has at most two child nodes (left and right). It is widely used in search trees, expression trees, and hierarchical data representations.

Binary Tree Representation of Expression: The given mathematical expression:

$$Y = \frac{(A + B)}{(C - D)^2}$$

can be represented in a Binary Expression Tree where: - Operators (+, -, /, ^) act as internal nodes. - Operands (A, B, C, D) act as leaf nodes.



💡 Quick Tip

Expression trees follow operator precedence, where operations with higher precedence (exponents, division) are placed higher in the tree.

(b) Explain in short, any four ways by which system security can be attacked.

Solution:

System security attacks involve unauthorized access, data breaches, and system disruptions. Here are four common types of attacks:

1. Phishing Attacks - Attackers trick users into providing sensitive information via fake emails or websites. - Often used to steal passwords, credit card numbers, and personal details.
2. Malware (Viruses, Worms, Trojans) - Malicious software that disrupts, damages, or gains unauthorized access to systems. - Can spread through email attachments, infected websites, or software downloads.
3. Denial of Service (DoS) Attacks - Attackers overload a system by flooding it with excessive requests, making it unavailable. - Common in large-scale website crashes or server slowdowns.
4. SQL Injection Attacks - Attackers inject malicious SQL queries into web applications to access or delete database information. - Can be prevented by using prepared statements and input validation.

💡 Quick Tip

Always follow cybersecurity best practices, such as firewalls, strong passwords, two-factor authentication, and regular system updates, to protect against attacks.

4(A). Answer any two of the following:

(a) What is a Class? Explain the general form of class declaration.

Solution:

Definition of a Class: A Class in C++ is a blueprint or a user-defined data type that allows grouping of data members (variables) and member functions (methods) together. It provides the foundation for object-oriented programming (OOP).

General Form of Class Declaration: A class is defined using the 'class' keyword followed by the class name and its body enclosed in curly braces “.

Syntax of Class Declaration:

```
class ClassName {  
    private:    //Privatemembers  
    protected: //Protectedmembers  
    public:     //Publicmembers  
        void display() {  
            cout << "Hello World";  
        }  
};
```

Example:

```
class Student {  
    public:  
        string name;  
        int age;  
        void display() {  
            cout << "Student Name:  " << name;  
        }  
};
```

Quick Tip

A class is a template for creating objects. Objects are instances of a class, and they help in encapsulation and data abstraction.

(b) What is Partitioning? Explain fixed and variable partitioning.

Solution:

Definition of Partitioning: Partitioning in operating systems refers to the division of memory into sections for efficient allocation and management of processes.

Types of Partitioning:

1. Fixed Partitioning:

- Divides memory into fixed-sized partitions.
- Each partition can hold only one process.
- Disadvantage: Leads to internal fragmentation (unused space within a partition).

2. Variable Partitioning:

- Divides memory dynamically based on process size.
- No fixed partitions; memory is allocated as needed.
- Advantage: Reduces internal fragmentation, making memory utilization efficient.

Example:

Type	Description
Fixed Partitioning	Memory divided into equal blocks (e.g., 4KB each)
Variable Partitioning	Memory allocated based on process size

 Quick Tip

Fixed partitioning is simple but inefficient due to wasted space, whereas variable partitioning optimizes memory utilization.

(c). Write the function of each of the following file stream classes:

(i) ifstream

Solution:

The ifstream (input file stream) class in C++ is used for reading data from a file. It allows input operations from files similar to reading data from the keyboard using 'cin'.

Example:

```
include <fstream >
using namespace std;
int main() {
    ifstream file("example.txt");
    string data;
    getline(file, data);
    cout << data;
    file.close();
}
```

 Quick Tip

Use ifstream when you only need to read from a file. Always close the file using file.close().

(ii) ofstream

Solution:

The ofstream (output file stream) class in C++ is used for writing data to a file. It allows output operations to files similar to writing data to the console using 'cout'.

Example:

```
include <fstream >
using namespace std;
int main() {
    ofstream file("example.txt");
    file << "Hello, File!";
    file.close();
}
```

💡 Quick Tip

Use `ofstream` when you only need to write to a file. Opening a file with `ofstream` erases existing content unless opened in append mode.

(iii) **filebuf**

Solution:

The `filebuf` (file buffer) class in C++ is used for low-level file handling by managing buffers for file input and output operations.

Example:

```
include <fstream >
using namespace std;
int main() {
    filebuf fb;
    fb.open("example.txt", ios::out);
    ostream os(&fb);
    os << "Using filebuf!";
    fb.close();
}
```

💡 Quick Tip

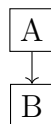
`filebuf` is rarely used directly but is useful when handling complex file streams or buffer manipulation.

4(B). Answer any one of the following:

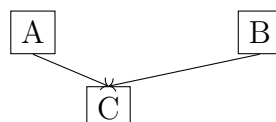
article tikz

(a) What is Inheritance? Identify the types of inheritance for the following diagrams:

(i)



(ii)



(iii)



Solution:

Definition of Inheritance: Inheritance is an Object-Oriented Programming (OOP) concept that allows one class (derived class) to acquire properties and behavior from another class (base class). It supports code reusability and establishes a hierarchical relationship between classes.

Identification of Inheritance Types:

(i) Single Inheritance:



- Description: - In single inheritance, a child class (B) inherits from a single parent class (A).
- The derived class can use properties and methods of the base class.
- Example in C++:

```

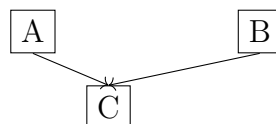
class A {
    public:
        void display() { cout << "Base Class"; }
};

class B : public A {
    public:
        void show() { cout << "Derived Class"; }
};
  
```

💡 Quick Tip

Single inheritance is the simplest form where a derived class inherits properties from a single parent class.

(ii) Multiple Inheritance:



- Description: - In multiple inheritance, a derived class (C) inherits from two or more parent classes (A, B).
- The child class gains access to features of multiple classes.

- Example in C++:

```
class A {
    public:
        void displayA() { cout << "Class A"; }
};

class B {
    public:
        void displayB() { cout << "Class B"; }
};

class C : public A, public B {
    public:
        void show() { cout << "Class C"; }
};
```

💡 Quick Tip

Multiple inheritance allows a class to inherit from multiple base classes, but it can lead to the diamond problem.

(iii) Multilevel Inheritance:

A
↓
B
↓
C

- Description: - In multilevel inheritance, a class (C) is derived from another derived class (B), which in turn is derived from (A). - This forms a chain of inheritance.

- Example in C++:

```
class A {
    public:
        void displayA() { cout << "Class A"; }
};

class B : public A {
    public:
        void displayB() { cout << "Class B"; }
};

class C : public B {
    public:
        void show() { cout << "Class C"; }
};
```

 Quick Tip

Multilevel inheritance helps in building hierarchical relationships, but deep inheritance chains can make code complex.

(b) What is Virtual Memory? Explain the following terms in virtual memory:

Solution: Definition of Virtual Memory: Virtual memory is a memory management technique that allows a computer to use disk space as an extension of RAM when physical memory (RAM) is insufficient. It helps in running large programs by storing parts of the program in secondary storage and loading them into RAM when needed.

(1) Page Fault

Solution: - A page fault occurs when a program accesses a page in virtual memory that is not currently loaded into RAM.

- The operating system must fetch the required page from secondary storage (hard disk) into RAM.
- Page faults slow down system performance but are managed using page replacement algorithms like FIFO, LRU, and Optimal Replacement.

 Quick Tip

Frequent page faults can lead to thrashing, where excessive swapping between RAM and disk slows down system performance.

(2) Dirty Page

Solution: - A dirty page is a page in RAM that has been modified but not yet written back to the disk.

- When a dirty page needs to be replaced, the operating system must write it to disk before removing it from RAM to prevent data loss.
- Dirty pages are tracked using page tables and memory management units (MMU).

 Quick Tip

Efficient handling of dirty pages improves system performance by reducing unnecessary disk writes.

5. Answer any two of the following:

(a) Write a program in C++ to find the Greatest Common Divisor (GCD) of two inputted numbers.

Solution:

The Greatest Common Divisor (GCD) of two numbers is the largest integer that divides both numbers exactly.

Here is the C++ program to find the GCD using the Euclidean algorithm:

```

include <iostream>
using namespace std;

// Function to find GCD using Euclidean Algorithm
int gcd(int a, int b) {
    while (b != 0) {
        int temp = b;
        b = a % b;
        a = temp;
    }
    return a;
}

int main() {
    int num1, num2;
    cout << "Enter two numbers: ";
    cin >> num1 >> num2;

    cout << "GCD of " << num1 << " and " << num2 << " is " << gcd(num1, num2) << endl;
    return 0;
}

```

Quick Tip

The Euclidean Algorithm is the most efficient way to compute GCD. It uses repeated division until the remainder is zero.

(b) Write a program in C++ to accept 10 integers in an array from the keyboard and find the largest element in the array.

Solution:

The following C++ program takes 10 integer inputs, stores them in an array, and finds the largest element.

```

include <iostream>
using namespace std;

int main() {
    int arr[10], maxElement;

    // Taking input for the array
    cout << "Enter 10 integers: ";
    for (int i = 0; i < 10; i++) {
        cin >> arr[i];
    }

    // Finding the largest element

```

```

maxElement = arr[0]; // Assume first element is the largest
for (int i = 1; i < 10; i++) {
    if (arr[i] > maxElement) {
        maxElement = arr[i];
    }
}

cout << "Largest element in the array is: " << maxElement << endl;
return 0;
}

```

💡 Quick Tip

To find the largest number in an unsorted array, iterate through the array while tracking the maximum value.

(c) Write an HTML program for the following table:

IND v/s ENG	
IND 387	Yuvraj 183*
	Dhoni 83
ENG 238	Peterson 58
	Bopara 49
IND WIN by 149 Runs	

Solution:

Below is the HTML code for creating the table structure exactly as shown in the question.

```

<!DOCTYPE html>
<html>
<head>
    <title>IND vs ENG Score Table</title>
    <style>
        table {
            width: 50%;
            border-collapse: collapse;
            text-align: center;
            font-family: Arial, sans-serif;
        }
        th, td {
            border: 2px solid black;
            padding: 10px;
        }
        th {

```

```

        font-size: 18px;
    }
</style>
</head>
<body>

<h2>IND v/s ENG</h2>

<table>
    <tr>
        <th colspan="2">IND v/s ENG</th>
    </tr>
    <tr>
        <td rowspan="2">IND 387</td>
        <td>Yuvraj 183</td>
    </tr>
    <tr>
        <td>Dhoni 83</td>
    </tr>
    <tr>
        <td rowspan="2">ENG 238</td>
        <td>Peterson 58</td>
    </tr>
    <tr>
        <td>Bopara 49</td>
    </tr>
    <tr>
        <td colspan="2"><strong>IND WIN by 149 Runs</strong></td>
    </tr>
</table>

</body>
</html>

```

💡 Quick Tip

Use colspan for merging columns and rowspan for merging rows in HTML tables. In LaTeX, the multicolumn and multirow commands help achieve the same effect.

OR

5. Answer any two of the following:

(a) Write a program in C++ to create a class test having member functions get-marks() to read marks of two subjects and showsum() to display the total marks.

Solution: Here is a C++ program that implements a class with member functions to handle marks calculation:

```
include <iostream>
using namespace std;
class Test {
private:
float subject1;
float subject2;
public:
// Function to get marks for two subjects
void getmarks() {
cout << "Enter marks for subject 1: ";
cin >> subject1;
cout << "Enter marks for subject 2: ";
cin >> subject2;
}

// Function to display total marks
void showsum() {
    float total = subject1 + subject2;
    cout << "Total marks: " << total << endl;
}

};
int main() {
Test student;
student.getmarks();
student.showsum();
return 0;
}
```

 Quick Tip

The use of private data members and public member functions demonstrates proper encapsulation, a key principle of object-oriented programming.

(b) Write a program in C++ to read three integers and find smallest of these numbers.

Solution: Here is a C++ program that finds the smallest number among three integers:

```
include <iostream>
using namespace std;
int main() {
int num1, num2, num3;

// Input three numbers
```

```

cout << "Enter three integers: ";
cin >> num1 >> num2 >> num3;

// Find smallest using conditional operators
int smallest = num1;

if (num2 < smallest) {
    smallest = num2;
}
if (num3 < smallest) {
    smallest = num3;
}

cout << "Smallest number is: " << smallest << endl;
return 0;

}

```

💡 Quick Tip

This solution uses sequential if statements for comparison, but you could also use the built-in `min()` function from the C++ Standard Library for a more concise solution.

(c) Write HTML code to display a hierarchical list showing different subjects under Arts, Commerce, and Science streams using appropriate HTML list tags.

LIST tag in HTML

- Arts
 - 1. Hindi
 - 2. Marathi
- Commerce
 - Tally
 - Accounts
- Science
 - A. Physics
 - B. Chemistry

Solution: Here is the HTML code that demonstrates the use of nested lists to create the required hierarchical structure:

```

<!DOCTYPE html>
<html>
<head>
    <title>LIST tag in HTML</title>
</head>
<body>
    <h1>LIST tag in HTML</h1>

```

```

<ul>
  <li>Arts
    <ol>
      <li>Hindi</li>
      <li>Marathi</li>
    </ol>
  </li>

  <li>Commerce
    <ul style="list-style-type: square;">
      <li>Tally</li>
      <li>Accounts</li>
    </ul>
  </li>

  <li>Science
    <ol type="A">
      <li>Physics</li>
      <li>Chemistry</li>
    </ol>
  </li>
</ul>

</body>
</html>

```

💡 Quick Tip

HTML supports different list types: unordered lists (`<ul>`), ordered lists (`<ol>`), and description lists (`<dl>`). The type attribute can be used to change the list markers, and nested lists can create hierarchical structures.