



# Shri Vaishnav Vidyapeeth Vishwavidyalaya, Indore

## Shri Vaishnav Institute of Information Technology

1: Title:- Program to display the following output using single cout statement.

Maths=90,physics=74,chemistry=76

2: Outcome:- display the desired output using a single cout statement

3: Objectives

The objective of this program is to display the scores of three subjects (Maths, Physics, and Chemistry) using a single cout statement. The scores are provided as variables (maths, physics, and chemistry) and are concatenated with the corresponding subject labels using the << operator. Finally, the std::endl is used to end the line and ensure the output is displayed.

4. Nomenclature:

cout	Display output to the screen
endl	It is a manipulator

4.1 Solution:

To display the desired output using a single cout statement, you can make use of stream insertion (<<) operator to concatenate the text and variables.

In the program, we declare three variables maths, physics, and chemistry to store the respective scores. Then we use a single cout statement to display the output by concatenating the text and variables using the << operator. Finally, we end the line with std::endl to move the cursor to the next line.

4.2 Code/ Pseudo Code

```
#include <iostream>

int main() {    int maths
= 90;
    int physics = 74; .    int
chemistry = 76;

    std::cout << "Maths=" << maths << ",physics=" << physics << ",chemistry=" <<
    chemistry << std::endl;

    return 0;
}
```



## Shri Vaishnav Vidyapeeth Vishwavidyalaya, Indore

### Shri Vaishnav Institute of Information Technology

#### 4.3 Output

```
Maths=90, Physics=74, Chemistry=76

-----
Process exited after 0.02511 seconds with return value 0
Press any key to continue . . . |
```



## Shri Vaishnav Vidyapeeth Vishwavidyalaya, Indore

### Shri Vaishnav Institute of Information Technology

1. Title:- Write a program to read 2 numbers from the keyboard and display the larger value on the screen.
2. Outcome:- user enter two values and it shows the larger value
3. Objective:- The objective of the "Number Comparison Program" is to compare two numbers entered by the user and determine the larger value
4. Nomenclature

#### 4.1 Nomenclature

Number Comparison Program	The name given to the program.
Input	The numbers entered by the user through the keyboard.
number1	The variable that stores the value of the first input number.
number2	The variable that stores the value of the second input number.
Larger Value	The greater of the two input numbers.
if statement	A control structure used to compare the input numbers and determine the larger value.
std::cin	The input stream object used to read values from the keyboard.
std::cout	The output stream object used to display messages on the screen.
<< operator	The stream insertion operator used to concatenate values for output.
>> operator	The stream extraction operator used to extract values from input.

#### 4.2 solution

- a. We declare two integer variables number1 and number2 to store the input values.
- b. The program prompts the user to enter the first number and reads it from the keyboard using std::cin.
- c. Similarly, the program prompts the user to enter the second number and reads it from the keyboard using std::cin.
- d. The program then uses an if statement to compare the values of number1 and number2.
- e. If number1 is greater than number2, it displays number1 as the larger value.
- f. If number2 is greater than number1, it displays number2 as the larger value.



## Shri Vaishnav Vidyapeeth Vishwavidyalaya, Indore

### Shri Vaishnav Institute of Information Technology

- g. If both numbers are equal, it displays a message indicating that both numbers are equal. Finally, the program ends and returns 0.

#### 4.3 Assumptions:-

1. The program expects the user to enter valid integer values for both numbers. It assumes that the user will not input non-numeric characters or floating-point numbers.
2. The program assumes that the user will enter the numbers one at a time and press the Enter key after each input.
3. The program assumes that the user will enter the numbers correctly and will not input excessive whitespace or non-numeric characters within the input.
4. The program assumes that the user will provide two numbers for comparison. It does not handle the case when the user enters fewer or more than two numbers.

#### 4.4 code/pseudo code

```
#include <iostream>

using namespace std;

int main() {

    int num1, num2;

    cout << "Enter two numbers: ";

    cin >> num1 >> num2;

    if (num1 > num2) {

        cout << "The larger value is: " << num1 << endl;

    } else {

        cout << "The larger value is: " << num2 << endl;

    }

    return 0;

}
```



## Shri Vaishnav Vidyapeeth Vishwavidyalaya, Indore

### Shri Vaishnav Institute of Information Technology

Output

```
Enter two numbers: 5 6
The larger value is: 6

-----
Process exited after 13.03 seconds with return value 0
Press any key to continue . . . |
```

1. Title: Write a function using reference variables as arguments to swap the values of a pair of integers.

2. Outcome:

- Develop a function that swaps the values of a pair of integers using reference variables.
- Demonstrate the functionality by swapping two integers and displaying the results.

3. Objective:

- Understand the concept of reference variables and their usage as function arguments.
- Implement a function to swap the values of two integers.
- Verify the successful swapping of values using a test case.
- Enhance knowledge of C++ programming and pass-by-reference.

4. Nomenclature:

swapIntegers	Function name that swaps the values of two integers
a, b	Reference variables representing integers
temp	Temporary variable to store a value during swapping
num1, num2	Integers to be swapped



## Shri Vaishnav Vidyapeeth Vishwavidyalaya, Indore

### Shri Vaishnav Institute of Information Technology

#### 4.1 Solution:

- Declare a function `swapIntegers` that takes two reference variables as arguments.
- Use a temporary variable to store the value of the first reference variable.
- Assign the value of the second reference variable to the first reference variable.
- Assign the value of the temporary variable to the second reference variable.
- In the `main` function, declare two integers and initialize them with values.
- Display the values before swapping.
- Call the `swapIntegers` function, passing the integers as arguments.
- Display the values after swapping.

#### 4.2 Assumptions:

- The function assumes that the provided integers are valid and within the range of `int`.
- The function assumes that the order of the integers matters when swapping.
- The function does not handle the case when the same integer is passed as both arguments.

#### 5. Code :



## Shri Vaishnav Vidyapeeth Vishwavidyalaya, Indore

### Shri Vaishnav Institute of Information Technology

```
#include <iostream>

void swapIntegers(int& a, int& b) {
    int temp = a;    a = b;    b = temp;
}

int main() {    int
num1 = 10;    int
num2 = 20;

    std::cout << "Before swapping: num1 = " << num1 << ", num2 = " << num2 << std::endl;

    swapIntegers(num1, num2);

    std::cout << "After swapping: num1 = " << num1 << ", num2 = " << num2 << std::endl;

    return 0;
}
```

#### Output

```
Before swapping: num1 = 10, num2 = 20
After swapping: num1 = 20, num2 = 10

-----
Process exited after 0.02373 seconds with return value 0
Press any key to continue . . . |
```

1. Title: Macro to Obtain the Largest of 3 Numbers



# Shri Vaishnav Vidyapeeth Vishwavidyalaya, Indore

## Shri Vaishnav Institute of Information Technology

### 2. Outcome:

- Develop a macro that determines the largest of three numbers.
- Demonstrate the usage of the macro to obtain the largest number among the given set.
- Display the largest number on the screen.

### 3. Objective:

- Understand the concept of macros in C++.
- Implement a macro to determine the largest number among three inputs.
- Apply the macro to compare and identify the largest number.
- Enhance knowledge of C++ programming and preprocessor directives.

### 4. Nomenclature:

- Macro	MAX_OF_3
- a, b, c	Input variables representing the three numbers
- largest	Variable to store the largest number

#### 4.1 Solution:

- Define a macro using the `#define` preprocessor directive.
- Name the macro `MAX_OF_3`.
- Pass three arguments, a, b, and c, representing the numbers to be compared.
- Utilize a ternary operator to determine the largest number among the three.
- Return the largest number from the macro.

#### 4.2 Assumptions:

- The macro assumes that the provided numbers are of the same data type (e.g., int) and within its range.
- The macro assumes that the order of the numbers matters and the comparison is done strictly based on their values.

#### 4.3 Dependencies:

- This solution does not have any external dependencies.

### 5. Code:

```
#include <iostream>
```





## Shri Vaishnav Vidyapeeth Vishwavidyalaya, Indore

### Shri Vaishnav Institute of Information Technology

```
#define MAX_OF_3(a, b, c) ((a > b && a > c) ? a : (b > a && b > c) ? b : c)
```

```
int main() {    int
num1 = 10;    int
num2 = 20;    int
num3 = 15;

    int largest = MAX_OF_3(num1, num2, num3);

    std::cout << "The largest number is: " << largest << std::endl;

    return 0;
}
```

#### Output

```
D:\sem 2\oops\macro.exe
Enter three numbers:
2
3
4
Largest number: 4
-----
Process exited after 3.318 seconds with return value 0
Press any key to continue . . .
```

#### 6. Viva Questions:

- What is a macro in C++?
- How does the #define directive work in defining a macro?
- Explain the purpose of the MAX\_OF\_3 macro in this program.
- How does the ternary operator contribute to the macro's functionality?
- Are there any limitations or considerations to keep in mind when using macros?

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

Data members

2. Outcome:

- Implement a class to represent a bank account.
- Assign initial values, deposit and withdraw amounts, and display account information.
- Develop a main program to test the bank account management system.



## Shri Vaishnav Vidyapeeth Vishwavidyalaya, Indore

### Shri Vaishnav Institute of Information Technology

#### 3. Objective:

- Understand the concept of classes and objects in C++.
- Implement a class with data members and member functions to manage a bank account. - Perform operations such as assigning initial values, depositing and withdrawing amounts, and displaying account information.
- Enhance knowledge of C++ programming and object-oriented principles.

#### 4. Nomenclature:

BankAccount	Class representing a bank account
- depositorName	Name of the account depositor (string)
- accountNumber	Account number (integer)
- accountType	Type of account (string)
- balanceAmount	Balance amount in the account (double)

#### 4.1 Solution:

- Define the BankAccount class with private data members for depositorName, accountNumber, accountType, and balanceAmount.
- Implement member functions to assign initial values, deposit an amount, withdraw an amount after checking the balance, and display account information.
- In the main program, create an instance of the BankAccount class.
- Call the member functions to assign initial values, perform deposit and withdrawal operations, and display the account details.

#### 4.2 Assumptions:

- The bank account management system assumes that the provided inputs are valid and within the appropriate range.
- The assumption is made that the withdrawal amount will not exceed the available balance in the account.
- The program assumes that the user interacts with the system through the main program.

#### 4.3 Dependencies:

- This program does not have any external dependencies.

#### 5. Code:

```
#include <iostream>
#include <string>
```



## Shri Vaishnav Vidyapeeth Vishwavidyalaya, Indore

### Shri Vaishnav Institute of Information Technology

```
class BankAccount { private:
    std::string depositorName;
    int accountNumber;
    std::string accountType;
    double balanceAmount;

public:
    BankAccount() {
        depositorName = "";
        accountNumber = 0;
        accountType = "";
        balanceAmount = 0.0;
    }
```

```
    void assignInitialValues(const std::string& name, int accNumber, const std::string& type,
double balance) {
        depositorName = name;
        accountNumber = accNumber;
        accountType = type;    balanceAmount
= balance;
    }

    void depositAmount(double amount) {
        balanceAmount += amount;
    }

    void withdrawAmount(double amount) {
        if (amount <= balanceAmount) {
            balanceAmount -= amount;
        } else {
```



## Shri Vaishnav Vidyapeeth Vishwavidyalaya, Indore

### Shri Vaishnav Institute of Information Technology

```
        std::cout << "Insufficient balance. Unable to withdraw." << std::endl;
    }
}

void display() {    std::cout << "Name: " <<
depositorName << std::endl;    std::cout << "Balance: $" <<
balanceAmount << std::endl;
}
};

int main() {
    BankAccount account;

    account.assignInitialValues("John Doe", 12345, "Savings", 1000.0);
    account.display();

    account.depositAmount(500.0);
    account.display();

    account.withdrawAmount(200.0);
    account.display();

    account.withdrawAmount(2000.0);
    account.display();

    return 0; }
```

Output :

```
Deposit of $500 successful.
Withdrawal of $200 successful.
Depositor name: John Smith
Account number: 12345
Account type: Checking
Balance: $1300

-----
Process exited after 0.01828 seconds with return value 0
Press any key to continue . . .
```



## Shri Vaishnav Vidyapeeth Vishwavidyalaya, Indore

### Shri Vaishnav Institute of Information Technology

#### 6. Viva Questions:

- What is a class in object-oriented programming?
- What are the advantages of using classes to represent real-world entities? □  
How are data members and member functions defined in a class



## Shri Vaishnav Vidyapeeth Vishwavidyalaya, Indore

### Shri Vaishnav Institute of Information Technology

1. Title: Create two classes DM and DB which store the value of distances. DM stores distances in meters and centimeters and DB in feet and inches. Write a program that can read values for the class objects and add one object of DM with another object of DB. Use a friend function to carry out the addition operation. The object that stores the results may be a DM object or DB object, depending on the units in which the results are required. The display should be in the format of feet and inches or meters and centimeters depending on the object on display.

2. Outcome:

- Implement two classes, DM and DB, to store distances in different units.
- Develop a friend function to add distances between DM and DB objects.
- Display the result in the format of feet and inches or meters and centimeters, depending on the input objects.
- Test the program by adding a DM object and a DB object.

3. Objective:

- Understand the concept of classes and friend functions in C++.
- Implement classes to store distances in different units and perform conversions.
- Develop a friend function to add distances between different units and handle unit conversions.
- Display the result based on the input objects' units to provide accurate and consistent output.
- Enhance knowledge of C++ programming and object-oriented principles.

4. Nomenclature:

-DM	Class representing distances in meters and centimeters
- DB	Class representing distances in feet and inches
- meters	Distance in meters (integer)
- centimeters	Distance in centimeters (integer)
- feet	Distance in feet (integer)
- inches	Distance in inches (integer)

4.1 Solution:

- Define the DM class with meters and centimeters as private data members.
- Define the DB class with feet and inches as private data members.
- Implement constructors in both classes to assign initial values.
- Declare the 'addDistances' function as a friend function of both DM and DB classes. - Inside the 'addDistances' function, convert the DM distance to inches and add it to the DB inches.
- Perform unit conversions and store the result in a DB object.
- Display the result in the appropriate format based on the input objects' units.



## Shri Vaishnav Vidyapeeth Vishwavidyalaya, Indore

### Shri Vaishnav Institute of Information Technology

#### 4.2 Assumptions:

- The program assumes that distances are non-negative and within a reasonable range.
- The assumption is made that the conversion factor of 1 meter = 39.3701 inches is accurate for the conversion of DM to DB distances.
- The program assumes that the user interacts with the system through the main program.

#### 4.3 Dependencies:

- This program does not have any external dependencies.

#### 5. Code:

```
#include <iostream>
```

```
class DB; // Forward declaration of the DB class
class DM { private:
    int meters;
    int centimeters;
public:
    DM(int m = 0, int cm = 0) : meters(m), centimeters(cm) {}

    friend void addDistances(DM objDM, DB objDB);

    void display() {      std::cout << "Distance in meters and centimeters: " << meters << "m "
<< centimeters << "cm" << std::endl;
    }
};

class DB { private:
    int feet;
    int inches;
```

```
public:
```



## Shri Vaishnav Vidyapeeth Vishwavidyalaya, Indore

### Shri Vaishnav Institute of Information Technology

```
DB(int ft = 0, int in = 0) : feet(ft), inches(in) {}

friend void addDistances(DM objDM, DB objDB);

void display() {      std::cout << "Distance in feet and inches: " << feet << "ft " << inches <<
"in" << std::endl;
}
};

void addDistances(DM objDM, DB objDB) {
    // Convert DM to DB    int dmInches = (objDM.meters * 100 + objDM.centimeters) * 0.393701; //
1 meter = 39.3701 inches
    DB resultDB(0, objDB.inches + dmInches);

    // Add DB objects
```

#### Output

```
Result: Distance in feet and inches: 5ft 10in
```

#### Viva question

1. What are the classes used in this program and their respective purposes?
2. How are the data members and member functions organized in the DM and DB classes?
3. What is the purpose of the friend keyword in the addDistances function declaration?
4. How does the addDistances function perform the addition of distances between DM and DB objects?
5. What conversions are performed in the addDistances function to ensure consistent units for addition?
6. How is the result displayed in the appropriate format based on the input objects' units?
7. Can you explain the purpose of the display function in both the DM and DB classes?





# Shri Vaishnav Vidyapeeth Vishwavidyalaya, Indore

## Shri Vaishnav Institute of Information Technology

1: Title: Design a constructor for bank account class.

2 :Outcome:

- Understand the concept and implementation of constructors in C++ classes.
- Design and implement a constructor for a bank account class to initialize its data members.
- Create bank account objects with initial values using the constructor.

3: Objective:

In this assignment, you will design and implement a constructor for a bank account class. The constructor will allow you to initialize the data members of the class when creating bank account objects. You will learn how to define and use constructors to simplify object initialization and ensure consistent state.

4: Nomenclature:

-

BankAccount	The name of the bank account class.
depositorName	A string representing the name of the depositor.
accountNumber	A string representing the account number.
accountType	A string representing the type of account.
balanceAmount	A double representing the balance amount in the account

.

4.1: Solution:

1. Define a BankAccount class with the necessary data members and member functions.
2. Implement a constructor that accepts parameters to initialize the data members.
3. Inside the constructor, assign the parameter values to the respective data members.
4. Add any additional initialization or validation logic, if required.

4.2: Assumptions:

- The bank account class is specific to a single depositor.
- The account number is unique for each account.
- The account type and balance amount can be modified after object creation using appropriate member functions.

4.3: Dependencies:

- C++ programming language.



## Shri Vaishnav Vidyapeeth Vishwavidyalaya, Indore

### Shri Vaishnav Institute of Information Technology

5: Code: \_

```
#include <iostream>
#include <string>

class BankAccount { private:
    std::string depositorName;
    std::string accountNumber;
    std::string accountType;    double
    balanceAmount;

public:
    BankAccount(const std::string& name, const std::string& number, const std::string& type, double
    balance)
        : depositorName(name), accountNumber(number), accountType(type), balanceAmount(balance) {
        // Additional initialization or validation logic, if required
    }

    // Other member functions of the BankAccount class };

int main() {
    // Create a bank account object using the constructor
    BankAccount account("John Doe", "1234567890", "Savings", 5000.0);

    // Test other member functions, if implemented

    return 0;
}
```

Output



## Shri Vaishnav Vidyapeeth Vishwavidyalaya, Indore

### Shri Vaishnav Institute of Information Technology

#### Bank Account Details:

Depositor Name: John Doe

Account Number: 1234567890

Account Type: Savings

Balance Amount: \$5000.00

#### 6: Viva question

1. What is the purpose of a constructor in a class?
2. How does a constructor differ from a regular member function?
3. How do you define and declare a constructor in C++?
4. What are the advantages of using constructors to initialize object data members?
5. How can you initialize data members using a constructor?
6. Can a constructor have parameters? If so, how are they used to initialize data members?
7. How would you modify the constructor to include additional initialization or validation logic?
8. What is the role of the this pointer in the constructor?
9. Can you create multiple constructors for a class? If yes, how can they be differentiated?
10. How would you handle default values for constructor parameters in the bank account class?

1: Title:

UTKARSH PATHAK

22100BTCSE11830

Sem II



## Shri Vaishnav Vidyapeeth Vishwavidyalaya, Indore

### Shri Vaishnav Institute of Information Technology

A book shop maintains the inventory of books that are being sold at the shop. The list includes details such as author, title, price, publisher and stock position. Whenever a customer wants a book, the sales person inputs the title and author and the system searches the list and displays whether it is available or not. If it is not, an appropriate message is displayed. If it is, then the system displays the book details and requests for the number of copies required. If the requested copies book details and requests for the number of copies required. If the requested copies are available, the total cost of the requested copies is displayed; otherwise the message "Required copies not in stock" is displayed.

Design a system using a class called books with suitable member functions and Constructors. Use new operator in constructors to allocate memory space required.

#### 2: Outcome:

- Understand the concept of class and object in C++.
- Implement a book inventory system using classes and objects.
- Practice member functions, data encapsulation, and conditional statements.

3: Objective: In this assignment, you will design and implement a book inventory system using classes and objects in C++. The system will allow users to search for books based on the title and author, check their availability, display book details, and request copies if available. You will learn to create classes, define member functions, and use conditional statements to fulfill the given requirements.

#### 4: Nomenclature:

Book	A class representing a book with properties such as title, author, publisher, price, and stock.
BookShop	A class representing a book shop that maintains an inventory of books.
title	A string representing the title of a book.
author	A string representing the author of a book.
publisher	A string representing the publisher of a book.
price	A double representing the price of a book.
stock	An integer representing the number of copies available in the inventory.

#### 4.1 Solution:



## Shri Vaishnav Vidyapeeth Vishwavidyalaya, Indore

### Shri Vaishnav Institute of Information Technology

1. Define a Book class with private data members to store the book details and public member functions to access and manipulate the data.
2. Implement a constructor in the Book class to initialize the data members when creating book objects.
3. Add member functions to the Book class for displaying book details, checking availability, and requesting copies.
4. Define a BookShop class with a vector or array to store the books in its inventory.
5. Implement a method in the BookShop class to add books to the inventory.
6. Create a method in the BookShop class to search for books based on the title and author.
7. Inside the search method, iterate through the inventory and compare the title and author to find a matching book.
8. If the book is found, check its availability and display the details accordingly.
9. If the requested copies are available, calculate the total cost and update the stock.
10. Handle cases where the book is not found or the requested copies are not available.

#### 4.2: Assumptions:

- The book inventory system is specific to a single book shop.
- Each book has a unique title and author combination.
- User input for book searches and requests is not implemented in the given code.

#### 4.3: Dependencies:

- C++ programming language.

#### 5: Output

```
#include<iostream>
Using namespace std;

// Define the Book class class
Book { private:
    // Data members public:
    // Constructor

    // Member functions
};
```



## Shri Vaishnav Vidyapeeth Vishwavidyalaya, Indore

### Shri Vaishnav Institute of Information Technology

```
class BookShop { private:
    // Inventory data member

public:
    // Methods for adding books and searching inventory };

int main() {
    // Create a BookShop object

    // Add books to the inventory

    // Search for books and test functionality

    return 0;
}
```

#### Output

```
Title: Book 1
Author: Author 1
Price: $10.99
Stock: 5
Enter the number of copies required: 3
Total Cost: $32.97
```

#### 6: Viva Questions:

1. What is the purpose of classes and objects in C++?
2. How do you define a class and its data members in C++?



## Shri Vaishnav Vidyapeeth Vishwavidyalaya, Indore

### Shri Vaishnav Institute of Information Technology

3. What are the advantages of using classes and objects in programming?
4. How does encapsulation contribute to code organization and maintenance?
5. What is the significance of member functions in a class? How are they defined and used?
6. How can you initialize the data members of a class using a constructor? 7. Explain the concept of data encaps