VYSYA COLLEGE, SALEM-103

CLASS: I BCA **PROGRAMS – 1 TO 3**

SUBJECT: PRACTICAL: C++ PROGRAMMING **SUBJECT CODE: 22UCAP02**

EX. NO: 1 - TO DEMONSTRATE FUNCTION OVERLOADING, DEFAULT ARGUMENTS AND INLINE FUNCTION.

AIM :

To write a C++ program to demonstrate Function Overloading, Default Arguments, and Inline Functions.

PROCEDURE:

Step 1: Start the Program

Step 2: Include Necessary Header Files

Step 3: Declare Namespace

Step 4: Define Function Overloading

Step 5: Define Default Arguments

Step 6: Define Inline Function

Step 7: Start Main Function

Step 8: Call add Function with Float Parameters(a & b)

Step 9: Call add Function with Integer Parameters(a & b)

Step 10: Call multiply Function with Default Argument(a)

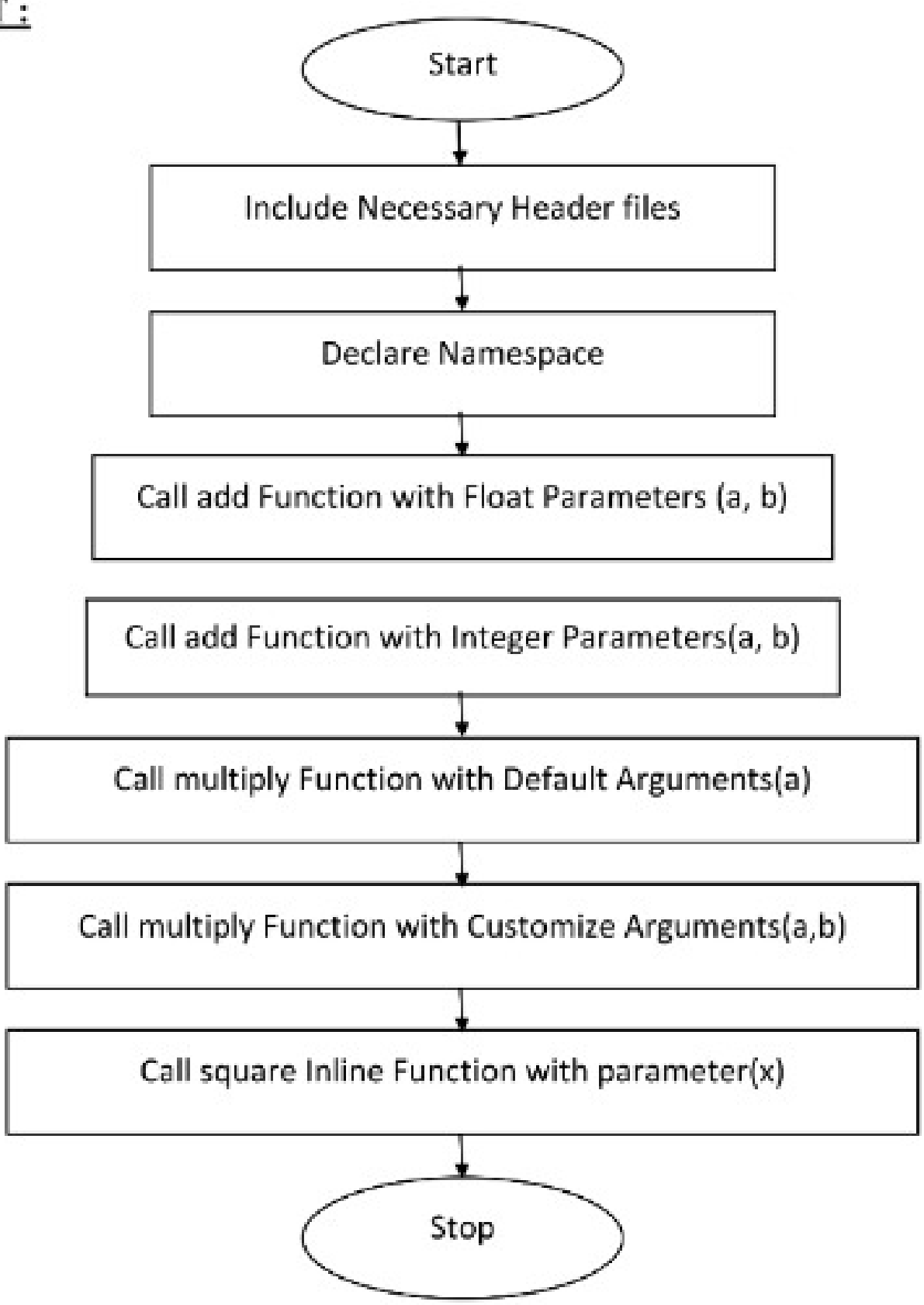
Step 11: Call multiply Function with Specified Arguments(a , b)

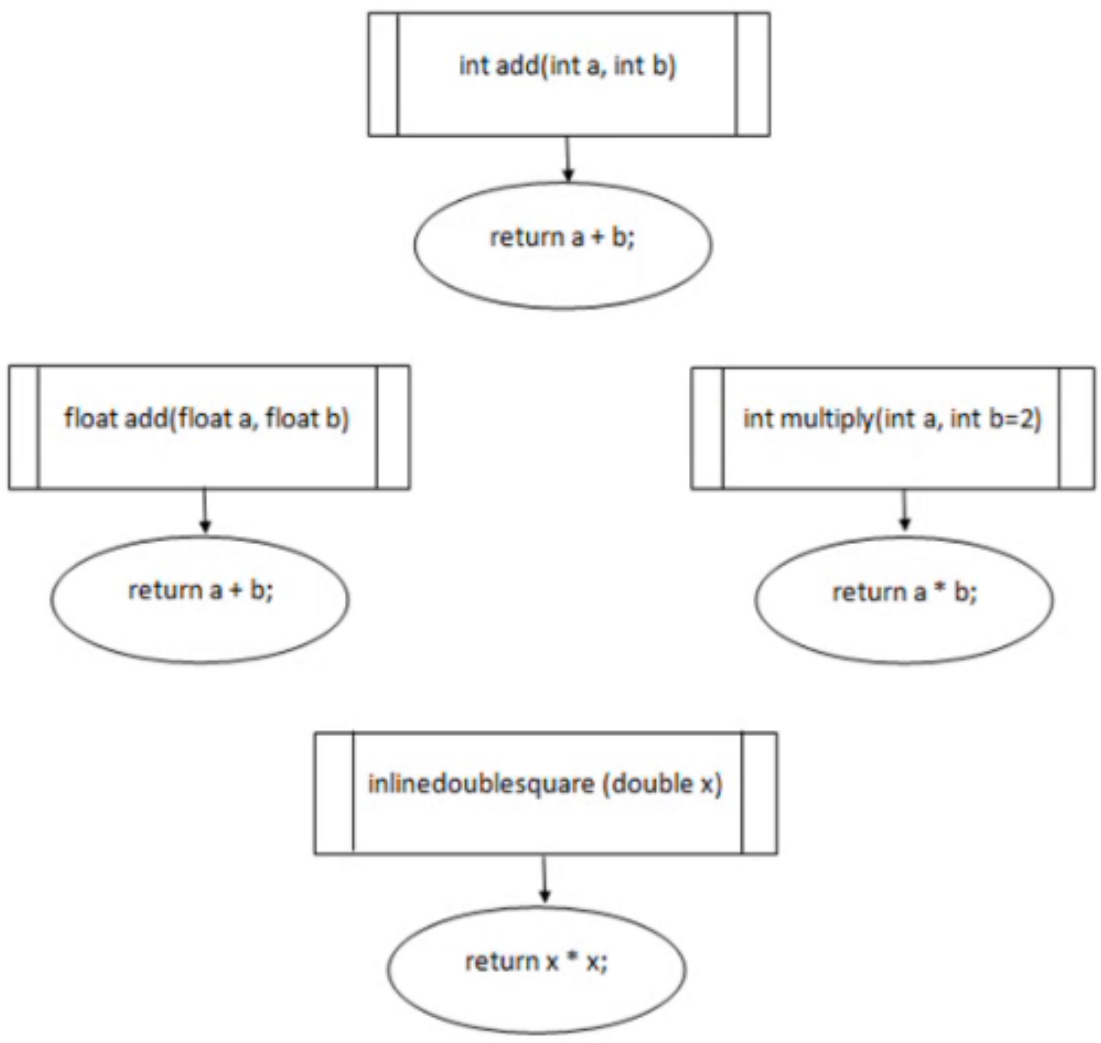
Step 12: Call square Inline Function with parameter (x)

Step 13: End of Main Function

Step 14 : Stop the program

FLOWCHART :


Start
Include Necessary Header files
Declare Namespace
Call add Function with Float Parameters (a, b)
Call add Function with Integer Parameters(a, b)
Call multiply Function with Default Arguments(a)
Call multiply Function with Customize Arguments(a,b)
Call square Inline Function with parameter(x)
Stop



SOURCE CODE :

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

// Function Overloading
int add(int a, int b)
{
        return a + b;
}
double  add(double  a, double  b)
{
        return a + b;
}
```

```
// Default Arguments
int multiply(int a, int b = 2)
{
        return a * b;
}

// Inline Function
inline double square(double x)
 {
return x * x;
}

int main()
{

 // Function Overloading
cout<<"Sum of doubles: "<< add(3.5, 2.7) <<endl;
cout<<"Sum of  integers: "<< add(5, 10) <<endl;

// Default Arguments
cout<<"Default multiplication: "<< multiply(4) <<endl;
cout<<"Custom multiplication: "<< multiply(4, 3) <<endl;

// Inline Function
double num = 4.0;
cout<<"Square of "<<num<<": "<< square(num) <<endl;
return 0;
}
```

Output:

Sum of Integers	:	15
Sum of Floats	:	6.2
Default Multiplication	:	8
Custom Multiplication	:	12
Square of 4	:	16

RESULT:

Thus , the demonstration of function overloading , default arguments and inline function concepts has been executed successfully

EX. NO: 2 - TO DEMONSTRATE CLASS & OBJECTS.

AIM :

To write a C++ program to demonstrate class and objects .

PROCEDURE :

Step1: Start the Program

Step2: Declare a Class "Room"

Step3: Define Public Data Members in the Class public: (length,breadth,height)

Step4: Define Member Functions for Calculating Area and Volume

Step5: End the Class Definition

Step6: Start the Main F unction

Step7: Create an Object (room1) of the " Room" Class

Step8: Assign Values to Data Members of the Object (40,30,20)

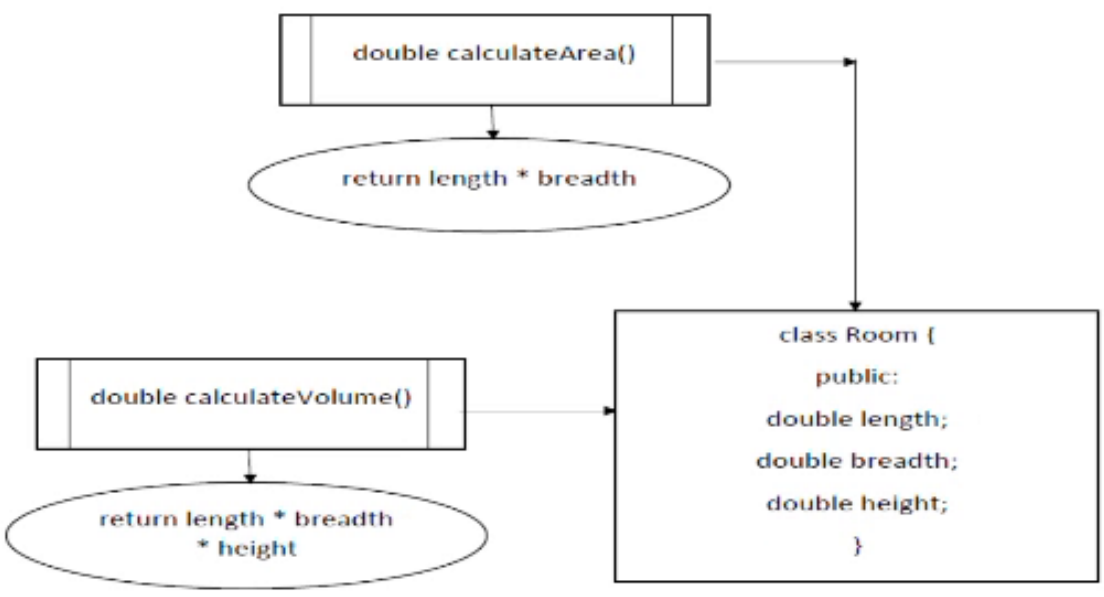
Step9: Calculate and Display the Area of the Room

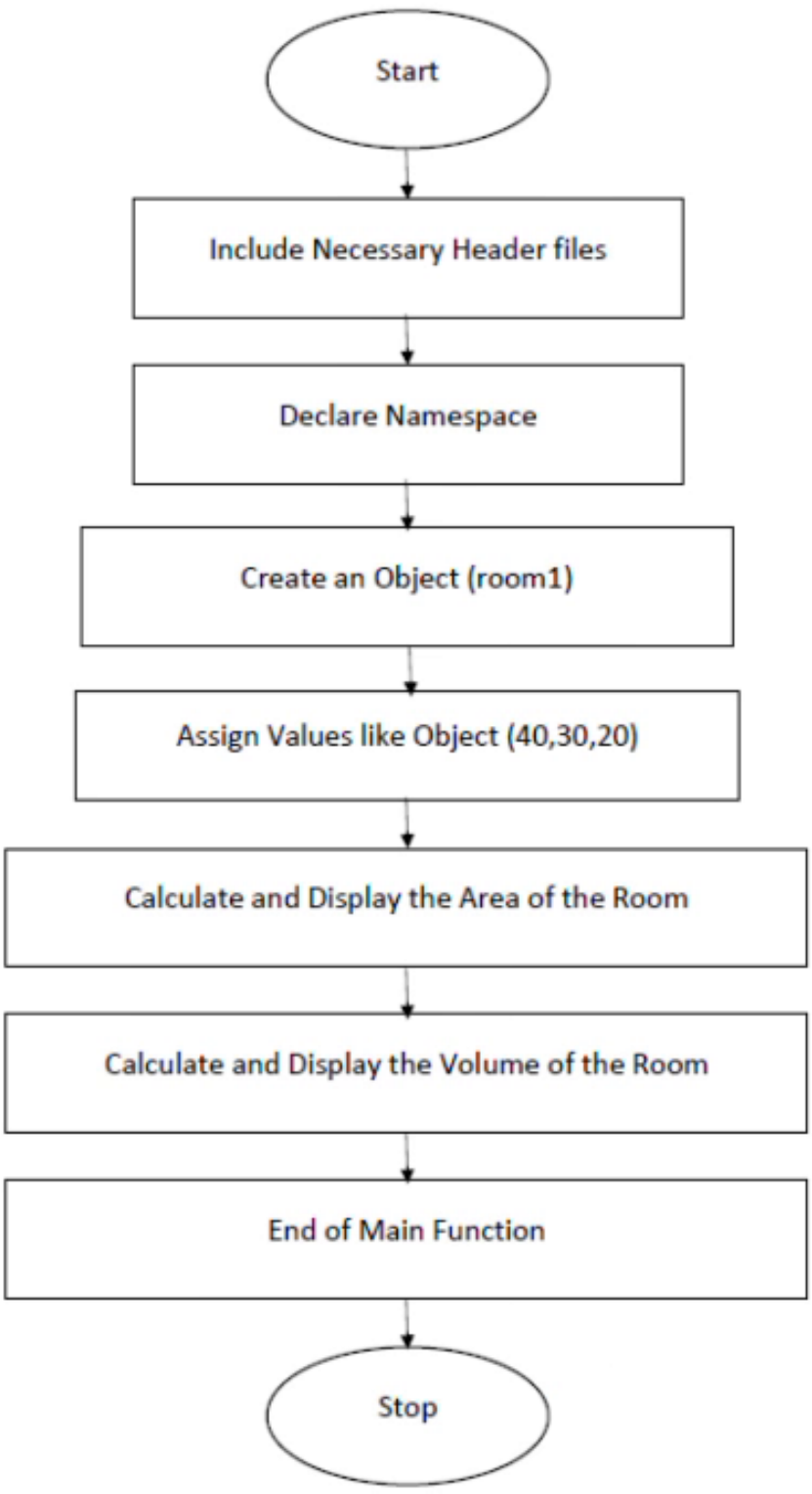
Step10: Calculate and Display the Volume of the Room

Step11: End the Main Function

Step12: Stop the program

FLOWCHART :





SOURCE CODE :

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

```
// create a class
class Room
{
public:
double length;
double breadth;
double height;
double calculateArea()
  {
       return length * breadth;
  }
double calculateVolume()
 {
       return length * breadth * height;
  }
};

int main()
{
// create object of Room class
   Room room1;
// assign values to data members
   room1.length = 4;
   room1.breadth = 3;
   room1.height = 2;

 // calculate and display the area and volume of the room
cout<<"Area of Room =  "<< room1.calculateArea()<<endl;
cout<<"Volume of Room =  "<< room1.calculateVolume()<<endl;
return 0;
}
```

OUTPUT:

```
Area of Room =  12
Volume of Room =  24
```

RESULT:

Thus , the demonstration of class and objects using C++ program has been executed successfully.

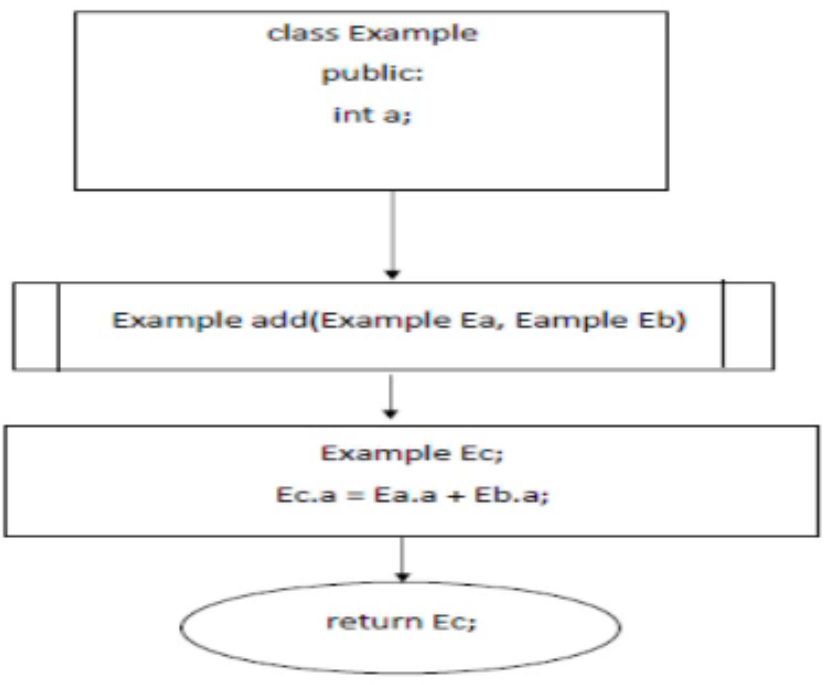
Ex. No : 3 - To Demonstrate the concept of Passing Objects to Functions

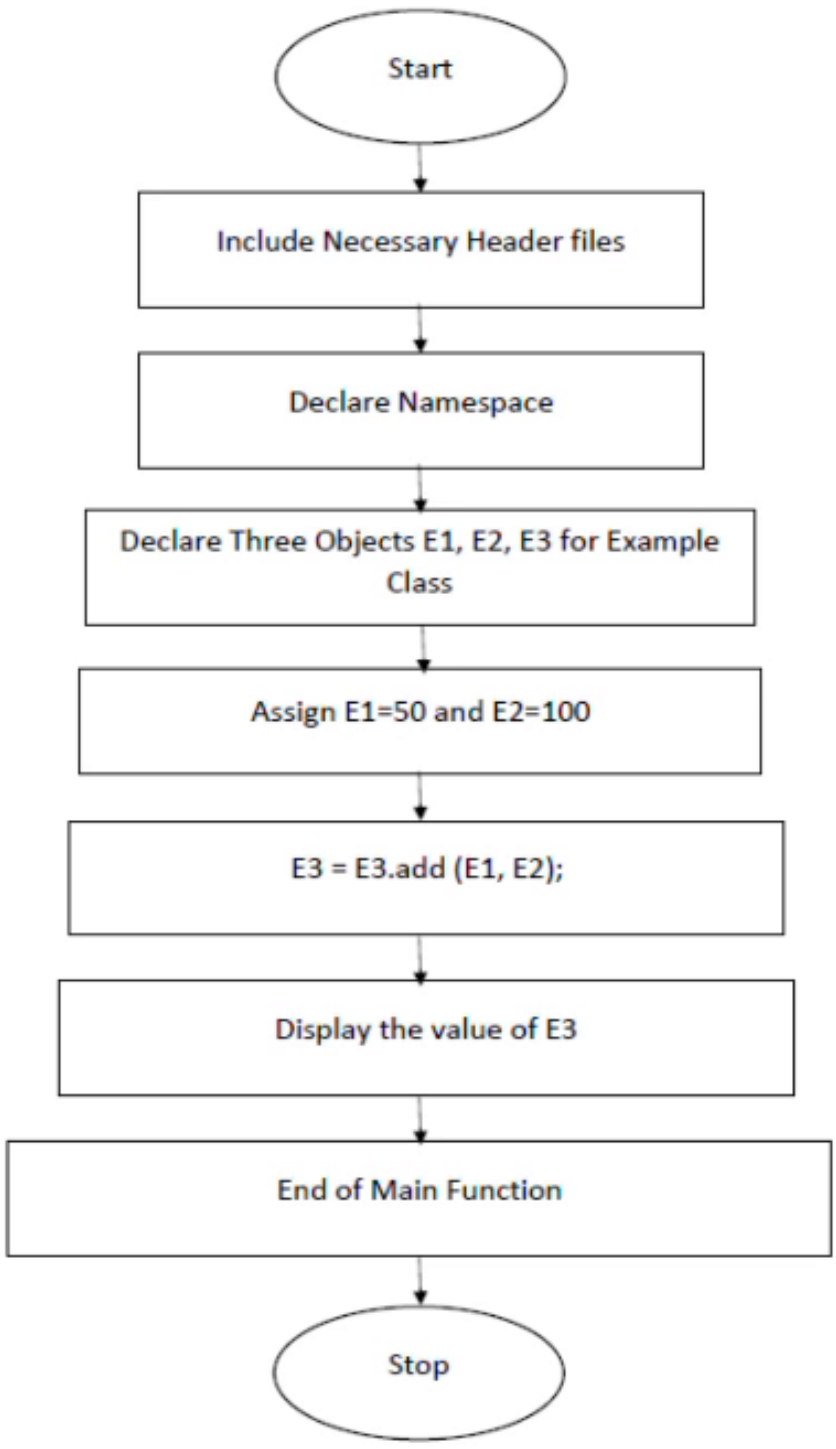
Aim : To write a C++ program to demonstrate class and objects with the concept of Passing Objects to Functions

Procedure :

Step 1: Start the Program
Step 2 : Include the necessary header file
Step 3 : Define a Class : EXAMPLE
Step 4 : Define Public Data Member variable(a) and member function (add) in the Class.
Step 5 : End the Class Definition
Step 6 : Start the Main Function
Step 7 : Declare Three objects E1,E2,and E3 for Example Class.
Step 8 : Initial values area signed for E1=50 and E2=100.
Step 9 : The add() is called with E1& E2 as arguments ,and the result is assigned to E3.
Step 10 : Finally, the updated value of E3 is displayed.
Step 11 : End the Main Function
Step 12 : Stop the program

FLOWCHART





SOURCE CODE :

```
#include<iostream>
using namespace std;
class Example
{
public:
int a;
// This function will take object as arguments and  return object
```

```
Example add(Example Ea, Example Eb)
{
Example Ec;
Ec.a = Ea.a + Eb.a;
// returning the object
return Ec;
}
};

int main()
{
Example E1, E2, E3;
// Values are initialized for both objects
E1.a = 50;
E2.a = 100;
E3.a = 0;
cout <<"Initial Values \n";
cout <<"Object 1: "<< E1.a<<endl;
cout <<"Object 2: "<< E2.a<<endl;
cout <<"Object 3: "<< E3.a<<endl;
// Passing object as an argument to function add()
E3 = E3.add(E1, E2);
// Changed values after passing object as an argument
cout <<"New values for Object 3 after Addition\n";
cout <<" Object 3: "<< E3.a;
return 0;
}
```

Output:

Initial Values
Object 1: 50
Object 2: 100
Object 3: 0
New values for Object 3 after Addition
Object 3: 150

RESULT:

Thus, the demonstration of passing objects to functions program was executed successfully.