VYSYA COLLEGE, SALEM-103

CLASS: I BCA **PROGRAMS – 4 TO 5**

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

EX. NO: 4 - TO DEMONSTRATE FRIEND FUNCTION

AIM:

To write a C++ program to demonstrate Friend Function.

PROCEDURE:

Step 1 : Start the program .

Step 2 : Includes the necessary header files.

Step 3 : Declare Friend Function named displayPrivateData outside the class.

Step 4 : Define the class MyClass with a private member privateData.

Step 5 : To Access the private member privateData of MyClass and display the result.

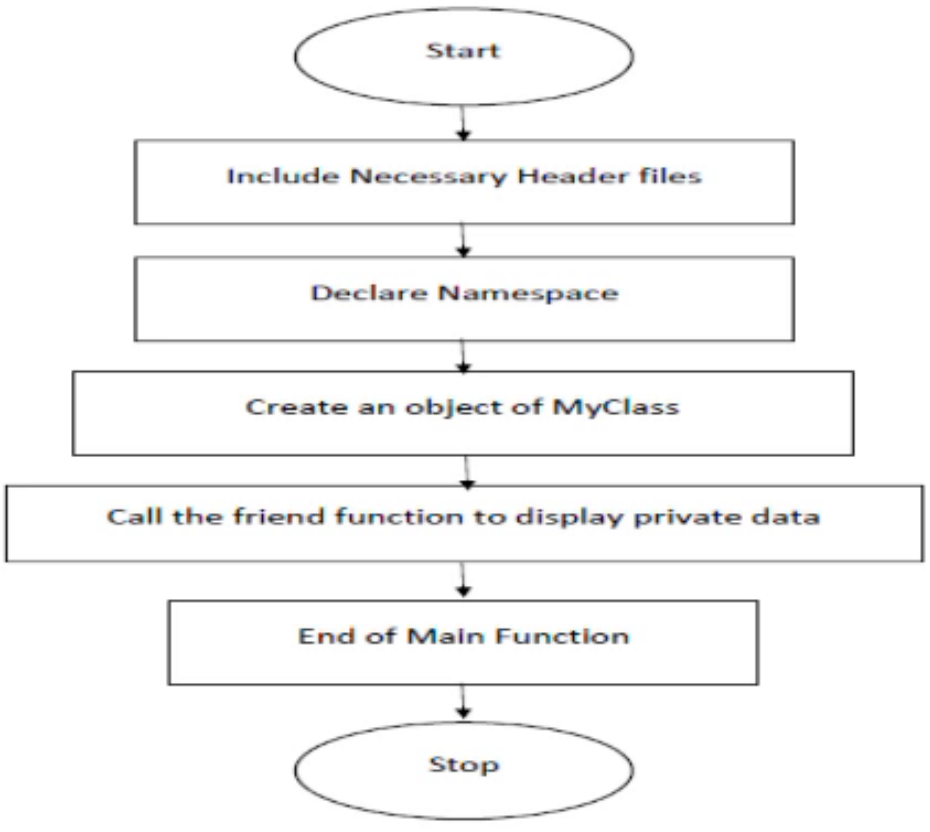
Step 6 : Define main function.

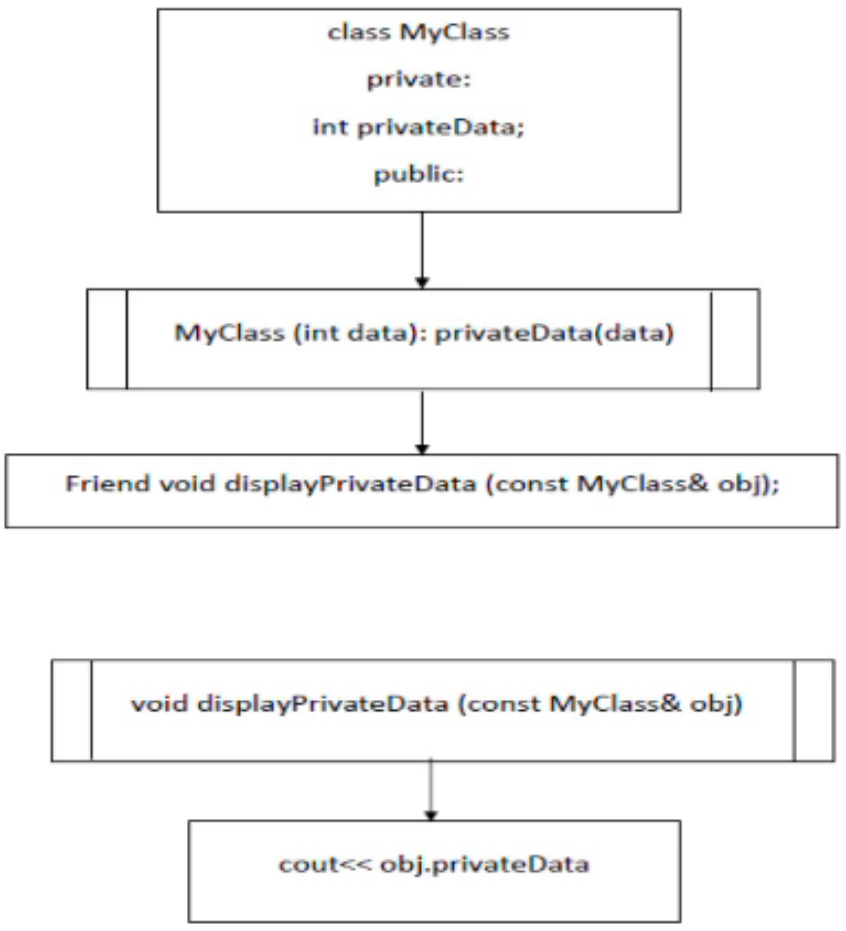
Step 7 : Create an object of MyClass named myObject.

Step 8 : Call the friend function with parameter (3337) to display private data .

Step 9 : Stop the program.

FLOWCHART





SOURCE CODE:

```
#include<iostream>
using namespace std;
// Forward declaration of the class
class MyClass;
// Friend function declaration
void displayPrivateData(const MyClass& obj);

// Class definition
class MyClass
{
private:
int privateData;
public:
 // Constructor
MyClass(int data) : privateData(data)
{
```

```
}
 // Friend function declaration
   friend void displayPrivateData(const MyClass& obj);
};

// Friend function definition
void displayPrivateData(const MyClass& obj)
{
   cout <<"Friend function accessing private data: "<< obj.privateData << endl;
}

int main()
{
 // Create an object of MyClass
   MyClass myObject(3337);
  // Call the friend function to display private data
   displayPrivateData(myObject);
  return 0;
}
```

OUTPUT:

Friend function accessing private data: 3337

RESULT:

Thus, the demonstration of friend function has been executed successfully executed.

Ex. No : 5 - 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.

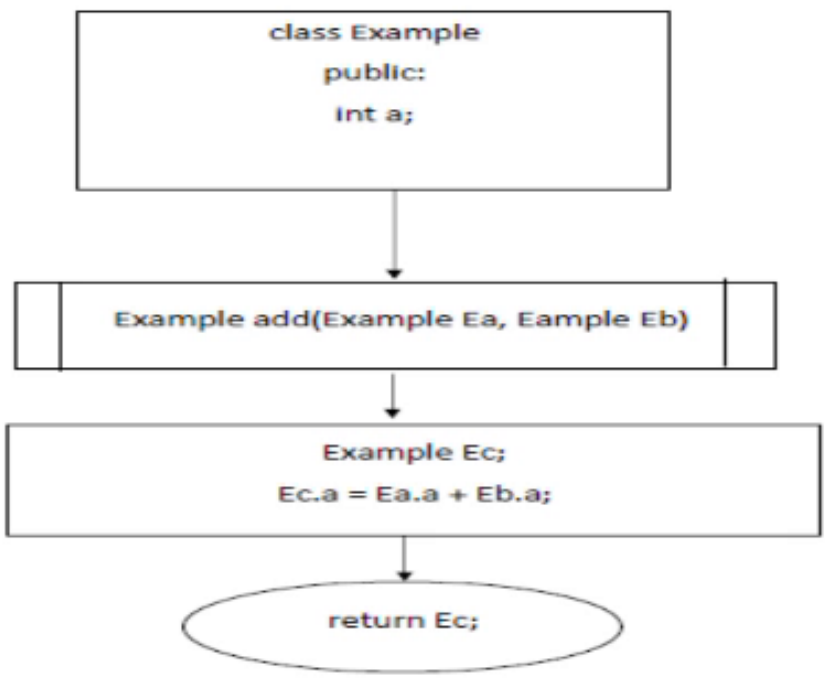
Step 9 : The add() is called with E1& E2 as arguments ,and the result is assigned to E3.

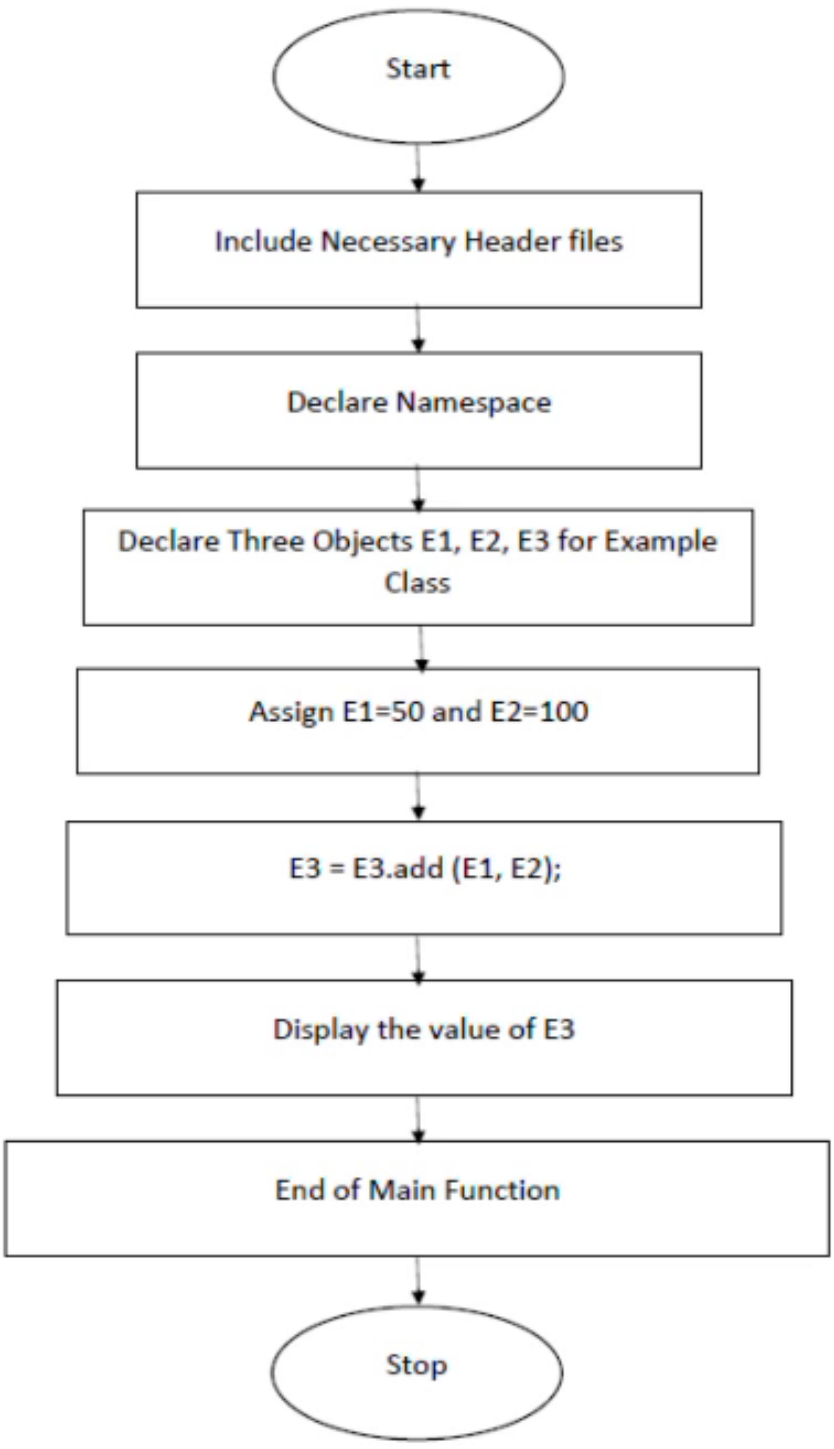
Step 10 : Finally, the updated value of E3 is displayed.

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.