

## Practical - 7

Ques. Write an algorithm, draw a flowchart and develop C++ program to demonstrate the static data member and static member function.

Algorithm :

Step 1 : Start.

Step 2 : Declare the test class.

Step 3 : Declare code as integer and static data member as a variable.

Step 4 : Define and declare the function setcode()

Step 5 : Use code = ++ count

Step 6 : Define and declare the function showcode for object member.

Step 7 : Create static member function showcount to get count.

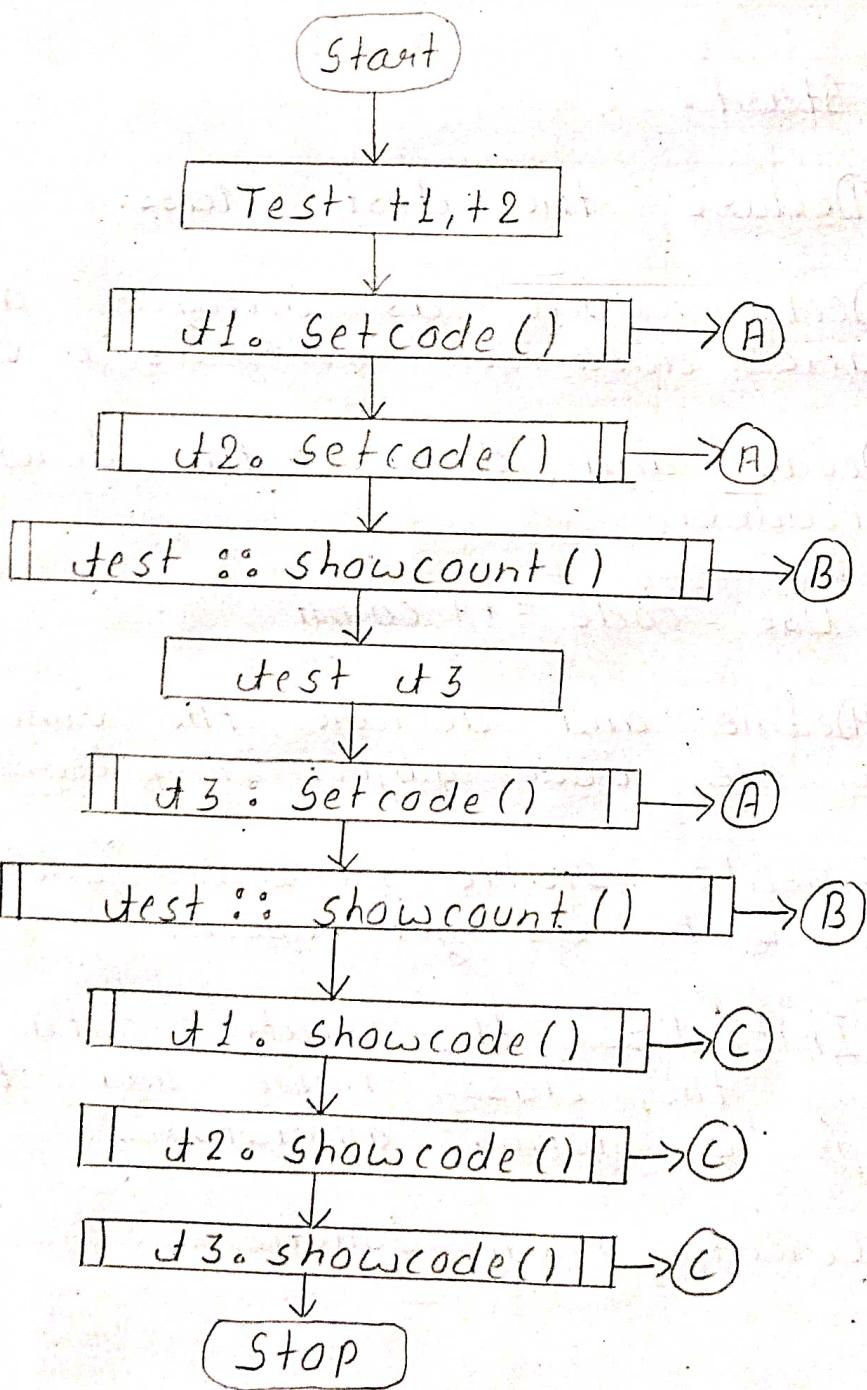
Step 8 : Initialize the static data member using the class name and the scope resolution operator.

Step 9 : Create main function.

# Flowchart:

①

```
class : test
static int count
int code
void setcode()
void showcode()
static void showcount()
```

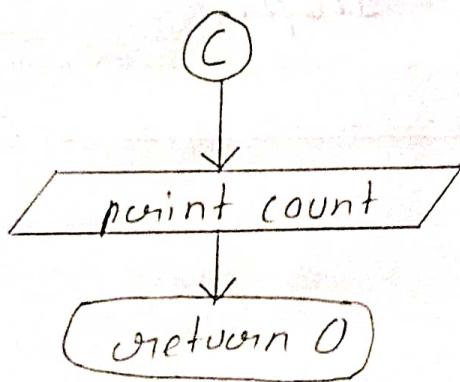
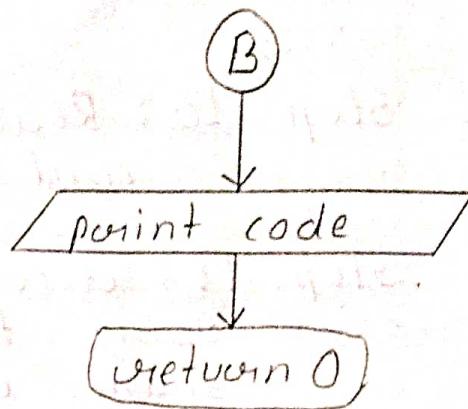
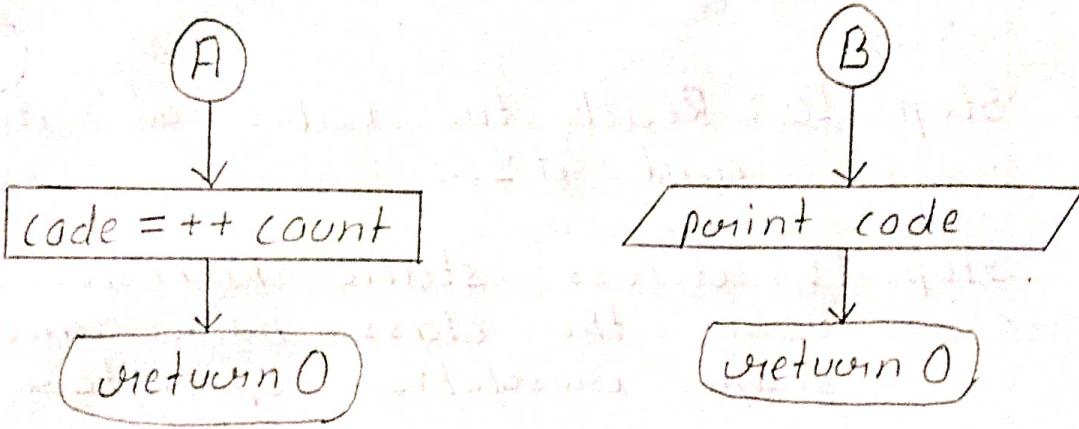


Step 10: Read the value of test t1 and t2.

Step 11: Access static members function using the class name and the scope resolution operator.

Step 12: Call the function setcode and showcade.

Step 13: Stop.



## PRACTICAL-7

Write an algorithm, draw a flowchart and develop a C++ to demonstrate the static data member and static member function.

### Program:

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

class test
{
    int code;
    static int count; // static member variable
public:
    void setcode()
    {
        code=++count;
    }
    void showcode(void)
    {
        cout<<"Object Member : "<<code<<endl;
    }
    static void showcount(void)
    {
        cout<<"Count="<<count<<endl;
    }
};

int test:: count;

int main()
{
    test t1,t2;
    t1.setcode();
    t2.setcode();
    test :: showcount(); // Count = 2
    test t3;
    t3.setcode();
    test:: showcount(); // Count = 3
    t1.showcode();
    t2.showcode();
    t3.showcode();
    return(0);
}
```

### Output:

```
D:\Bcca Ist yr\SEM II\C++ Programming\C++ practi... - □ X
Count=2
Count=3
Object Member : 1
Object Member : 2
Object Member : 3
-----
Process exited after 0.09013 seconds with return value 0
Press any key to continue . . .
```

## Practical - 8

Ques. Write an algorithm, draw a flowchart and develop a C++ program to demonstrate the use of single inheritance

Algorithm :

Step 1 : Start.

Step 2 : Declare the base class.

Step 3 : Define and declare the function `getdata()`.

Step 4 : Declare and derived class derive.

Step 5 : Declare and define the function `readdata()` to get the y details.

Step 6 : Define function `product()` to calculate x and y.

Step 7 : Create the derived class object.

Step 8 : Derive a.

Step 9 : Call the function `getdata()`, `readdata()`, `product()`.

Step 10 : Stop.

⑧

## Flowchart:

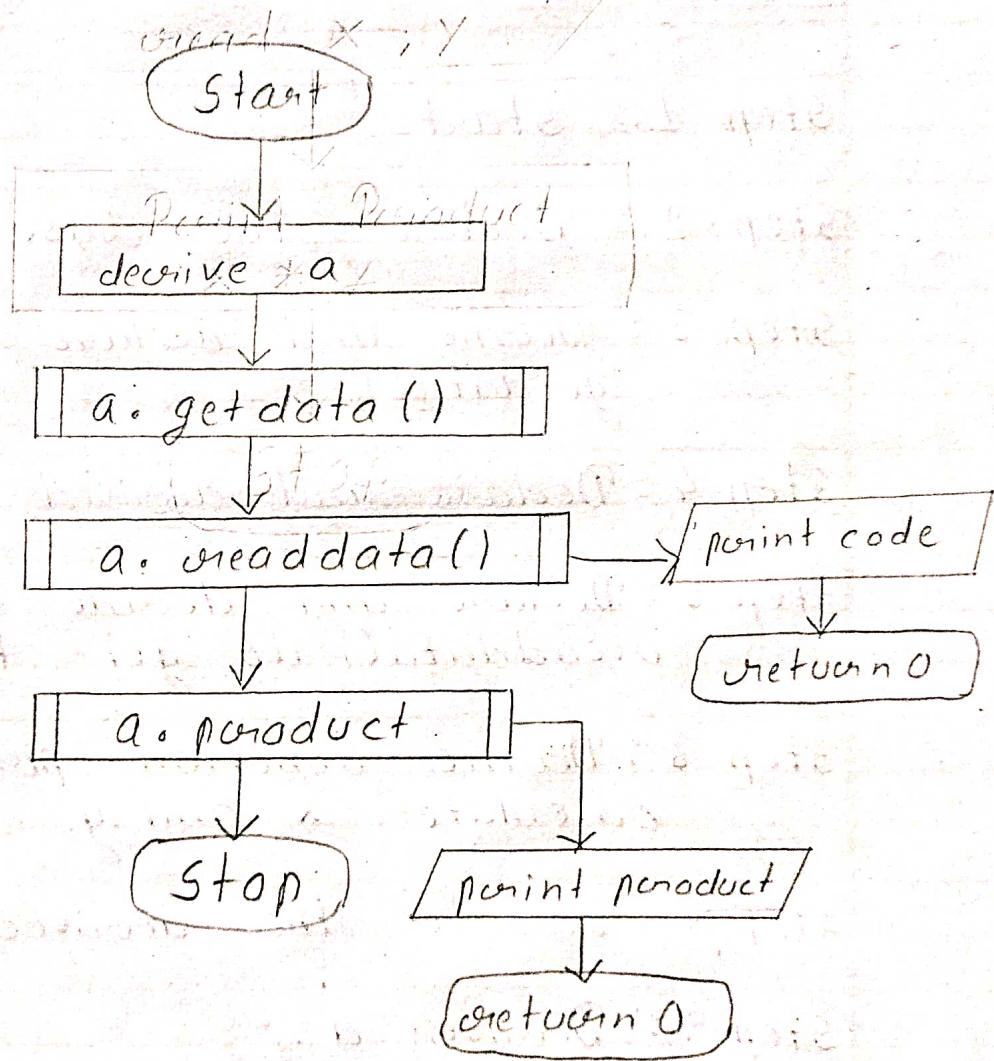
①

class : base

int x

void getData()  
void readData()  
void product()

②



## PRACTICAL-8

Write an algorithm, draw a flowchart and develop a C++ program to demonstrate the use of single inheritance

### Program:

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

class base
{
public:
    int x;
    void getdata()
    {
        cout << "Enter the value of x = ";
        cin >> x;
    }
};

class derive : public base
{
private:
    int y;
public:
    void readdata()
    {
        cout << "Enter the value of y = "; cin >> y;
    }

    void product()
    {
        cout << "Product = " << x * y;
    }
};

int main()
{
    derive a;
    a.getdata();
    a.readdata();
    a.product();
    return 0;
}
```

### Output:

```
Select D:\Vccs 1st yr\SEM 1\c++ Programming\c++
Enter the value of x = 13
Enter the value of y = 16
Product = 208
Process exited after 0.125 seconds with return value 0
Press any key to continue . . .
```

## Practical - 9

Ques. Write an algorithm, draw a flowchart and develop a C++ program to display Fibonacci series  
(i) using recursion, (ii) using iteration

Algorithm : using recursion

Step 1 : Start.

Step 2 : Declare fib and x as integers.

Step 3 : Set condition int fib function.

i. if ( $x == 1$ ) || ( $x == 0$ ) then return(x).

ii. else return (fib( $x - 1$ ) + fib( $x - 2$ )).

Step 4 : Declare and initialize variable x and i=0 in main function.

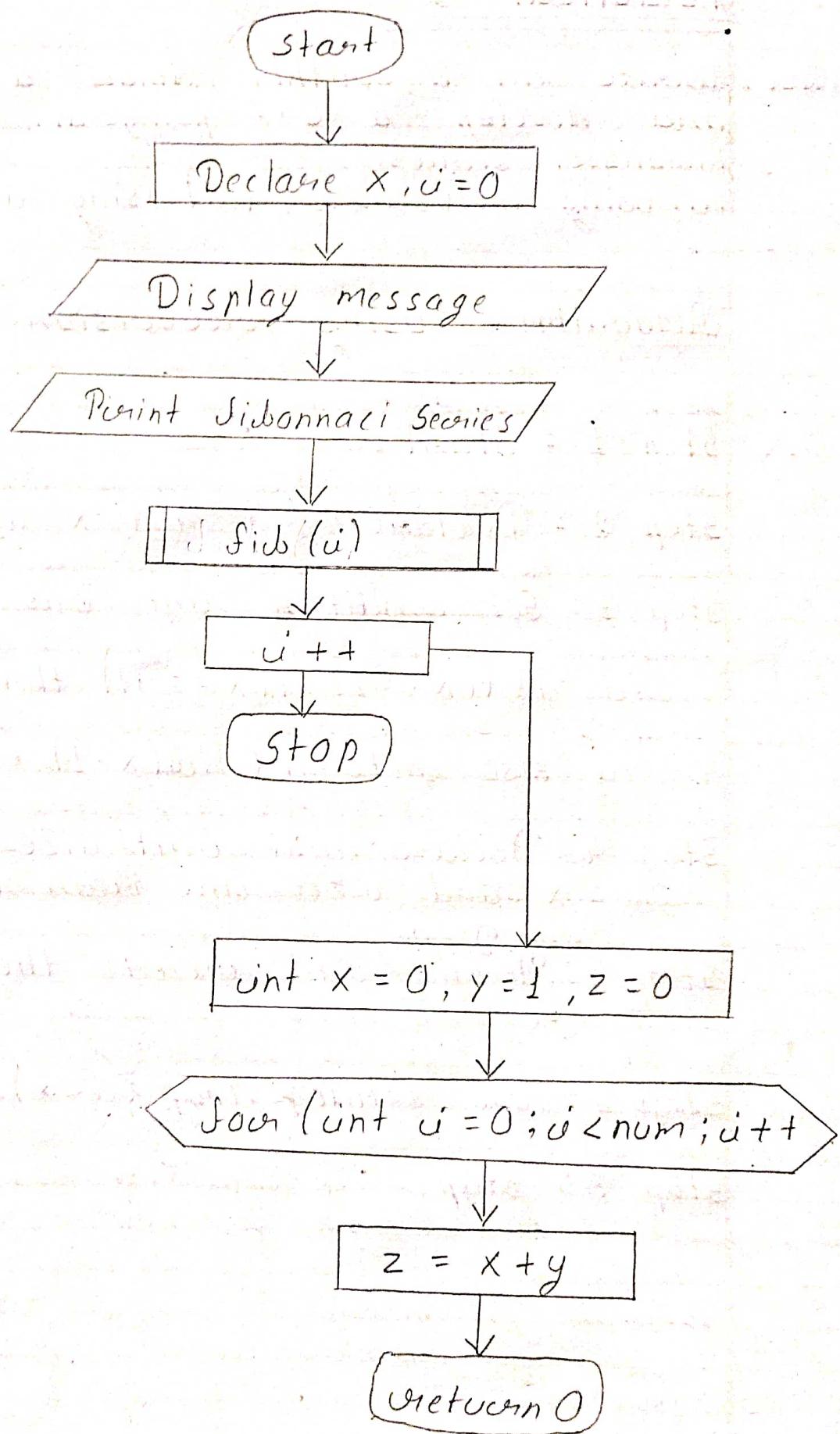
Step 5 : ~~Declare~~ <sup>Display</sup> and read the value of x.

Step 6 : Use while loop ( $i < x$ ).

Step 7 : Stop.

⑨

## Flowchart :



## PRACTICAL- 9

Write an algorithm, draw a flowchart and develop a C++ program to display Fibonacci series.

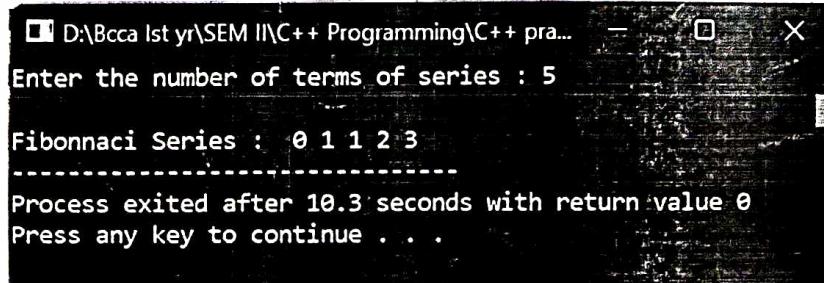
(i) using recursion:

**Program:**

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

int fib(int x)
{
    if((x==1)||(x==0))
    {
        return(x);
    }
    else
    {
        return(fib(x-1)+fib(x-2));
    }
}
int main()
{
    int x , i=0;
    cout << "Enter the number of terms of series : ";
    cin >> x;
    cout << "\nFibonacci Series : ";
    while(i < x)
    {
        cout << " " << fib(i);
        i++;
    }
    return 0;
}
```

**Output:**



D:\Bcca 1st yr\SEM II\C++ Programming\C++ pr...

Enter the number of terms of series : 5

Fibonacci Series : 0 1 1 2 3

Process exited after 10.3 seconds with return value 0

Press any key to continue . . .

### (iii) Using iteration

Algorithm : Using iteration

Step 1 : Start.

Step 2 : Declare num as integer.

Step 3 : Enter the the number from user.

Step 4 : Go to fib(num) function.

Step 5 : Initialize  $x=0$ ,  $y=1$  and  $z=0$ .

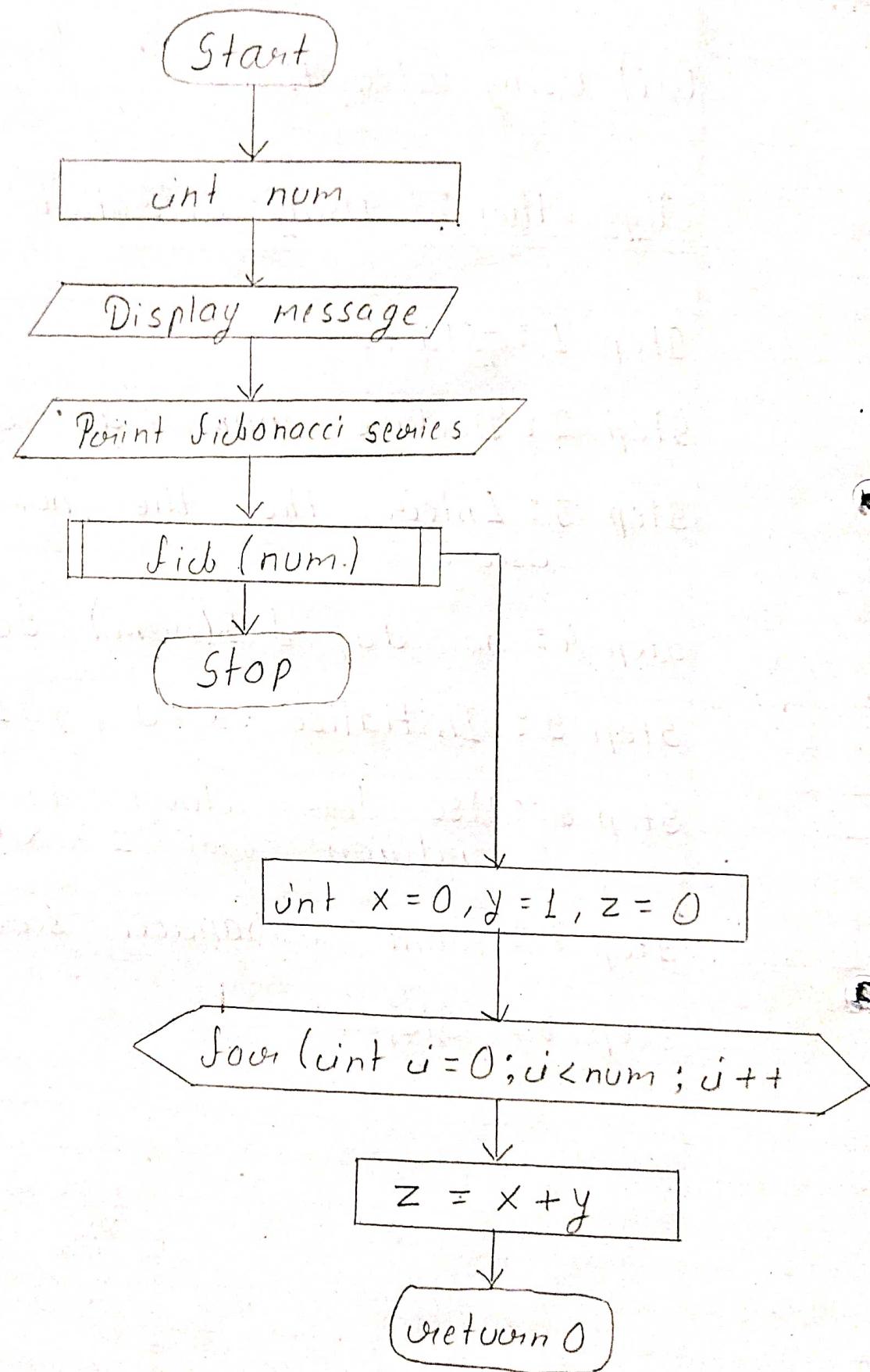
Step 6 : Use for loop to check condition for  $z = x + y$

Step 7 : Print Fibonacci series.

Step 8 : Stop.

⑨

## Flowchart:



## PRACTICAL-9

Write an algorithm, draw a flowchart and develop a C++ program to display Fibonacci series.

### (ii) using iteration:

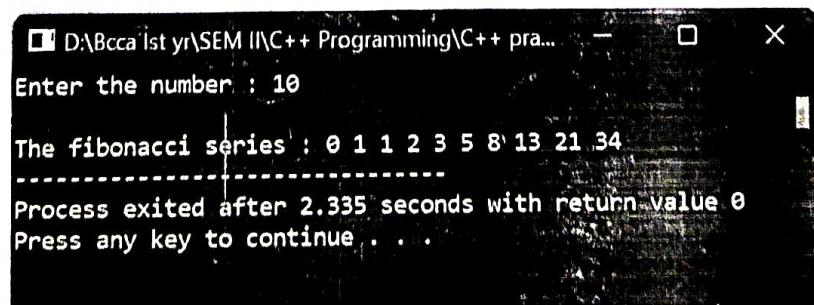
#### **Program:**

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

void fib(int num)
{
    int x = 0, y = 1, z = 0;
    for (int i = 0; i < num; i++)
    {
        cout << x << " ";
        z = x + y;
        x = y;
        y = z;
    }
}

int main()
{
    int num;
    cout << "Enter the number : ";
    cin >> num;
    cout << "\nThe fibonacci series : ";
    fib(num);
    return 0;
}
```

#### **Output:**



D:\Bcca 1st yr\SEM II\C++ Programming\C++ pr...

Enter the number : 10

The fibonacci series : 0 1 1 2 3 5 8 13 21 34

Process exited after 2.335 seconds with return value 0

Press any key to continue . . .

## Practical - 10

Que. Write an algorithm, draw a flowchart and develop a C++ program to create a class Triangle. Include overloaded functions for calculating area. Overload assignment operator and equality operator.

Algorithm :

Step 1: Start.

Step 2: Create class triangle.

Step 3: Declare area as float s1,  
float s2, float s3

Step 4: Initialize the of float  $s = s1 + s2 + s3$ ,  $s = s/2$  and  $x = \sqrt{(s)*(s-s1)*(s-s2)*(s-s3)}$

Step 5: print area by x.

Step 6: Initialize Declare area float ht  
and float cbt

Step 7: calculate height & base length

Use formula  $x = ht * cbt * (0.5)$

Step 8: Print area & stop.

⑩

## Flowchart:

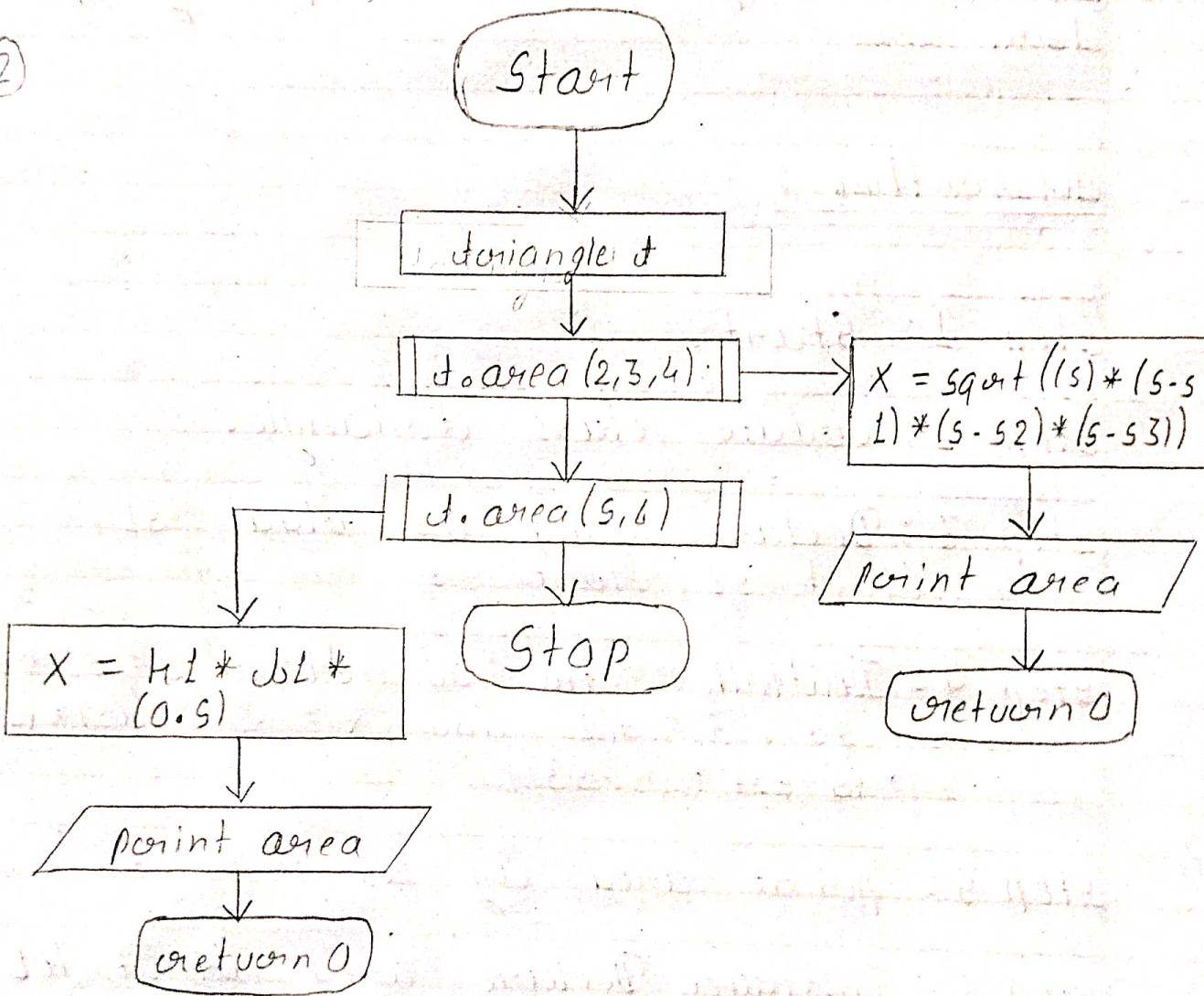
①

```
class : triangle
```

```
void area (float s1, float s2,
           float s3)
```

```
void area (float h1, float b1)
```

②



## PRACTICAL- 10

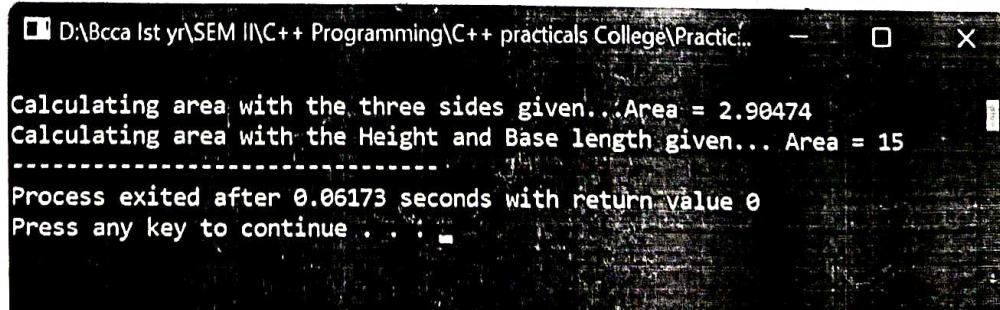
Write an algorithm, draw a flowchart and develop a C++ program to create a class Triangle. Include overloaded functions for calculating area. Overload assignment operator and equality operator.

### Program:

```
#include<iostream>
#include<math.h>
using namespace std;

class triangle // Creating class triangle
{
public:
void area(float s1,float s2,float s3)
{
    cout<<"\nCalculating area with the three sides given...";
    double X;
    float s=s1+s2+s3;
    s = s/2;
    X=sqrt((s)*(s-s1)*(s-s2)*(s-s3));
    cout<<"Area = "<<X;
}
void area(float h1,float b1)
{
    cout<<"\nCalculating area with the Height and Base length given... ";
    double X;
    X = h1*b1*(0.5);
    cout<<"Area = "<<X;
}
int main()
{
    triangle t;
    t.area(2,3,4);
    t.area(5,6);
    return 0;
}
```

### Output:



```
D:\Bcca 1st yr\SEM II\C++ Programming\C++ practicals College\Practic.. - X
Calculating area with the three sides given...Area = 2.90474
Calculating area with the Height and Base length given... Area = 15
Process exited after 0.06173 seconds with return value 0
Press any key to continue . . .
```

## Practical - 11

Ques. Write an algorithm, draw a flowchart & develop a C++ program to show the constructor overloading.

Algorithm :

Step 1 : Start.

Step 2 : Declare the person class.

Step 3 : Declare int age.

Step 4 : Constructor with no arguments.

Step 5 : Constructor with an arguments.

Step 6 : Display person age.

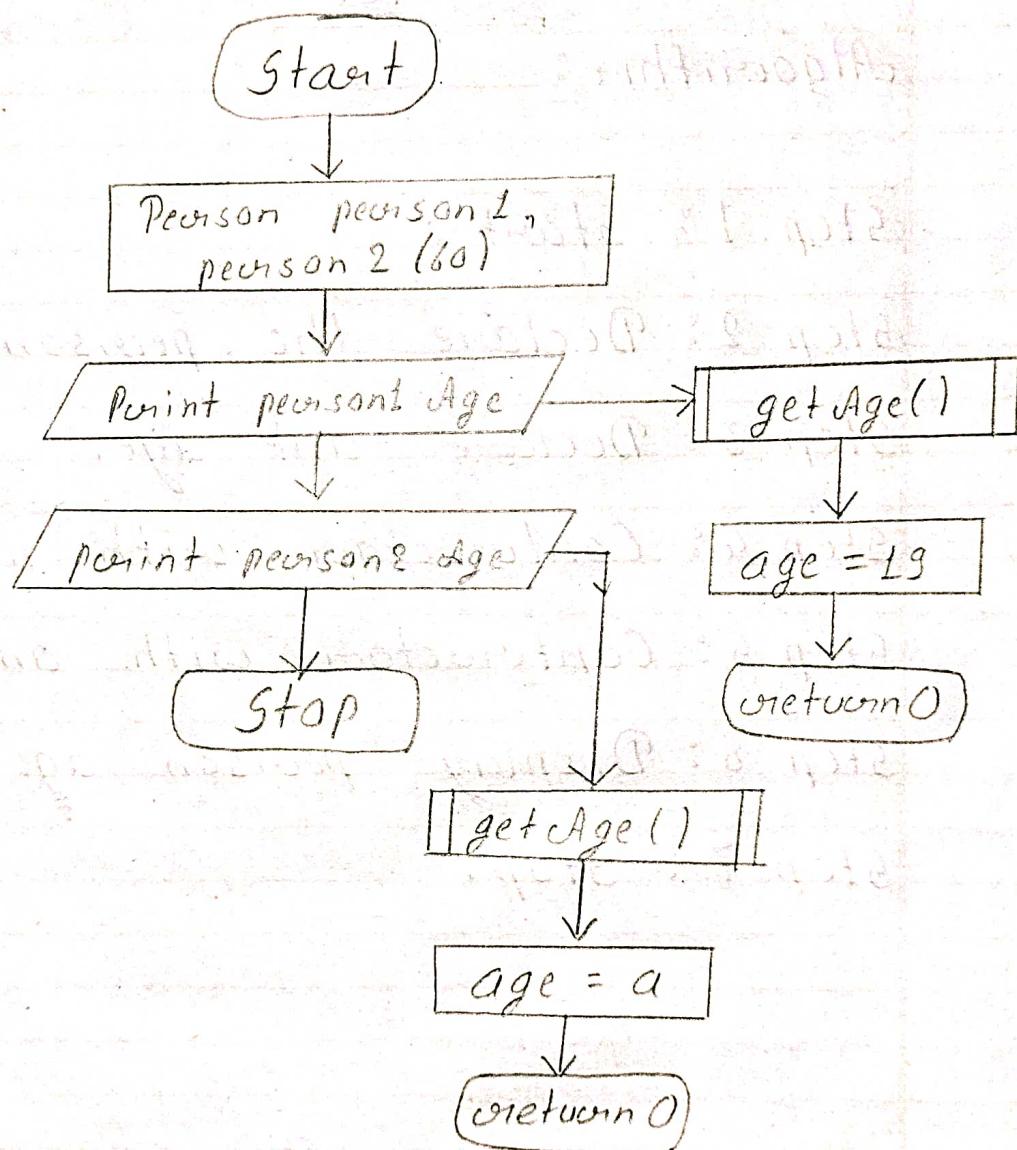
Step 7 : Stop. ~~person beyond~~

12

## Flowchart:

①

```
class : Person
int a
int getAge()
```



## PRACTICAL- 11

Write an algorithm, draw a flowchart and develop a C++ program to show the constructor overloading.

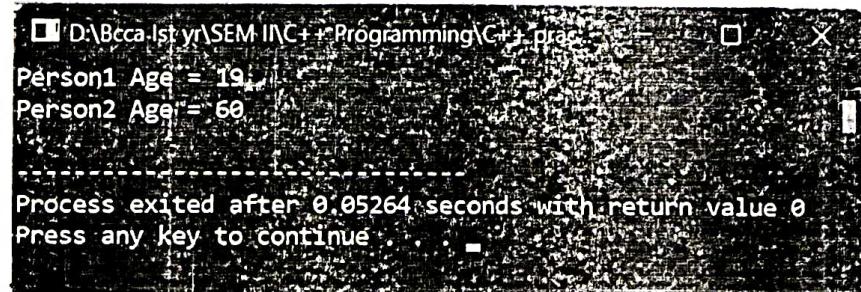
### **Program:**

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

class Person
{
private:
    int age;
public:
    Person() //Constructor with no arguments
    {
        age = 19;
    }
    Person(int a) //Constructor with an argument
    {
        age = a;
    }
    int getAge()
    {
        return age;
    }
};

int main()
{
    Person person1, person2(60);
    cout << "Person1 Age = " << person1.getAge() << endl;
    cout << "Person2 Age = " << person2.getAge() << endl;
    return 0;
}
```

### **Output:**



```
D:\Bca-1st yr\SEM II\C++ Programming\c++_prac
Person1 Age = 19
Person2 Age = 60
Process exited after 0.05264 seconds with return value 0
Press any key to continue...
```