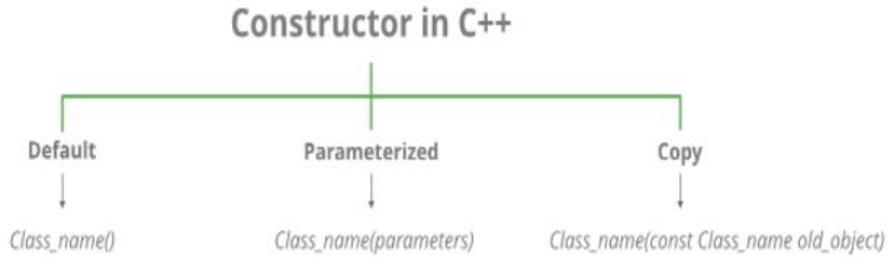
PP Lab Name: Shilpi Biswal
PRN:22070122198



Practical No 9 **Date:** /08/2022

Title: Write a C++ program to perform addition of two complex numbers using the concept of constructor overloading and friend function.

Description:

A constructor is a special method of a class or structure in object-oriented programming that initializes a newly created object of that type. Whenever an object is created, the constructor is called automatically.

A constructor is like an instance method that usually has the same name as the class, and can be used to set the values of the members of an object, either to default or to user-defined values. However, although it resembles it, a constructor is not a proper method since it doesn't have a return type. Instead of performing a task by executing code, the constructor initializes the object

The declaration: integer int1;
not only creates the object int1 of type integer but also initializes its data members m and n to zero. There is no need to write any statement to invoke constructor.

Hint:

Define three constructors:
First constructor with no arguments: For the objects which are not initialized
Second constructor: Takes one argument and initializes real and imag part to equal values
Third constructor: Takes two arguments and initializes real and imag part to two different values
)

```
class Complex
{
int real,imag;
public:
  Complex()                 //Default Constructor
  {

  }
Complex(int r)
```

PP Lab Name: Shilpi Biswal
PRN:22070122198

```
  //Parameterised constructor for equal values
  {
……
…….
  }
Complex(int r, int i)
 //Parameterised constructor for different values
  {
  ….
……
  }

  void print()
  {
…….
  }
friend Complex sum(Complex,Complex);   //create  the object and return it in the definition of friend function
};

int main()
{
Complex c1(10);
Complex c2(10,20);
sum(c1,c2);
return 0;
}
```

Program Code:

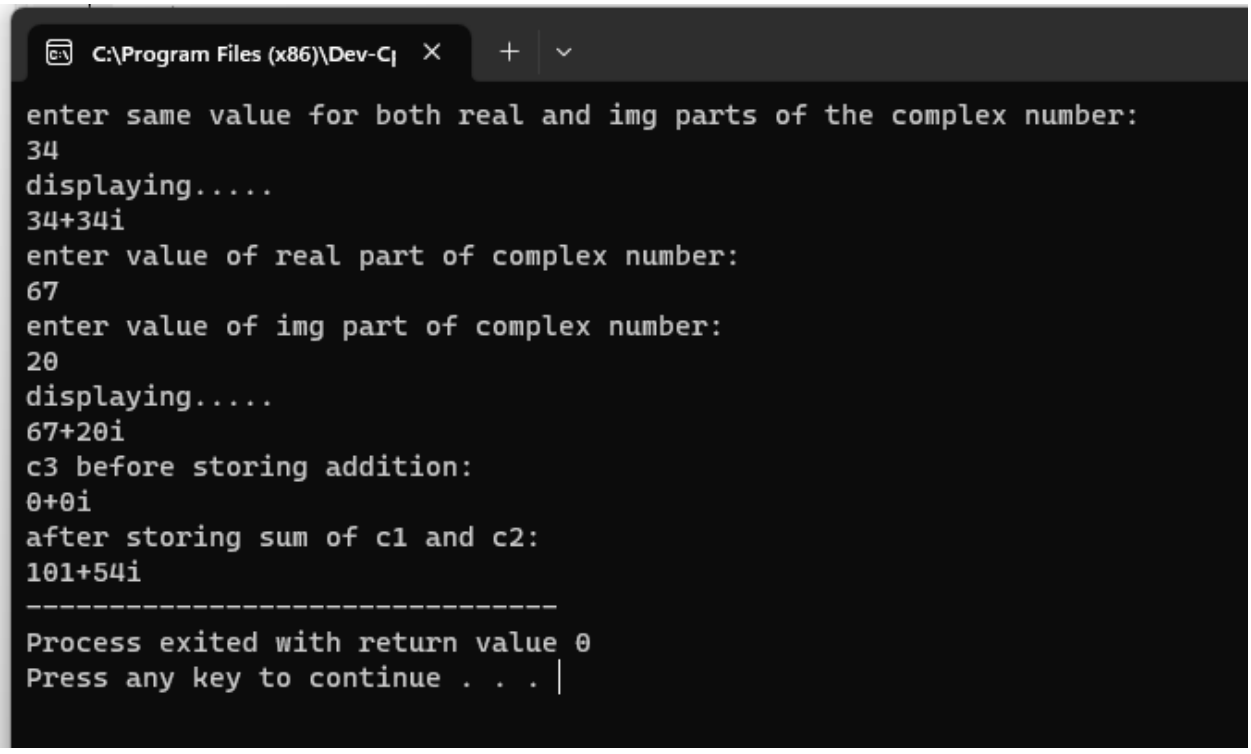
constructors.cpp series.cpp

```
#include<iostream>
using namespace std;
class Complex
{
    int real;
    int img;
public:
    Complex(){
        real=0;
        img=0;

    }
    Complex(int i)
    {
        real=i;
        img=i;
    }
    Complex(int a, int b)
    {
        real=a;
        img=b;
    }
    friend Complex add(Complex c1, Complex c2)
    {
        Complex result;
        result.real=c1.real+c2.real;
        result.img=c1.img+c2.img;
        return result;
    }
    void display()
    {
        cout<<real<<"+"<<img<<"i";
    }
};
```

PP Lab Name: Shilpi Biswal
PRN:22070122198

```
int main()
{
    int real;
    int img;
    cout<<"enter same value for both real and img parts of the complex number:"<<endl;
    cin>>real;
    Complex c1(real);
    cout<<"displaying....."<<endl;
    c1.display();
    cout<<"\nenter value of real part of complex number:"<<endl;
    cin>>real;
    cout<<"enter value of img part of complex number:"<<endl;
    cin>>img;
    Complex c2(real,img);
    cout<<"displaying....."<<endl;
    c2.display();
    cout<<"\nc3 before storing addition: "<<endl;
    Complex c3;
    c3.display();
    cout<<"\nafter storing sum of c1 and c2: "<<endl;
    c3=add(c1,c2);
    c3.display();

}
```

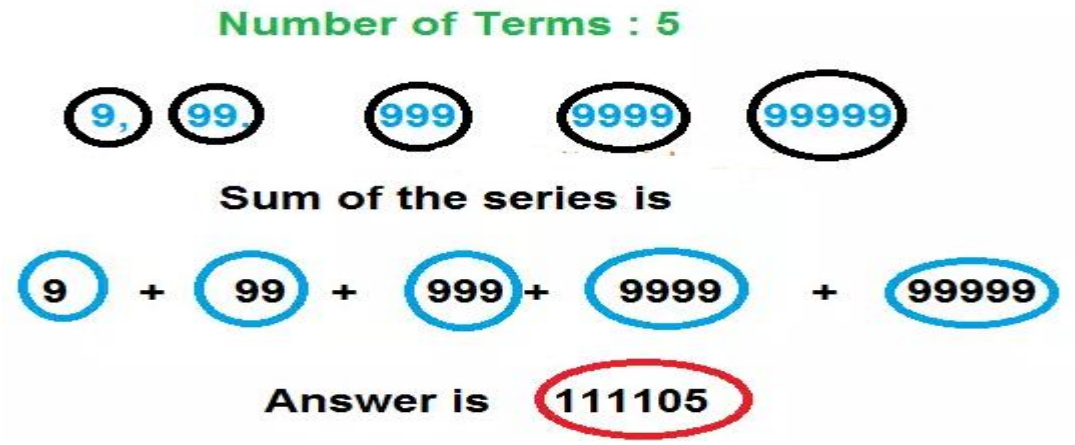
Input and Output



PP Lab Name: Shilpi Biswal
PRN:22070122198

Conclusion: Thus we have implemented the concept of constructors in C++

Practice programs: Write a program in to find the Sum of the series using Constructor Overloading in C++.



Class complex{

public:
complex(); //Constrctor
complex(int img, int real); //Constructor overloading

}

int main()
{
complex c1;
complex c2(10,12);

PP Lab Name: Shilpi Biswal
PRN:22070122198

Program Code:

constructors.cpp series.cpp series2.cpp

```cpp
#include<iostream>
using namespace std;
class Series
{
    int sum;
public:
    Series(){

    }
    Series(int n){
        sum=0;
        int term;

        for(int i=1;i<=n;i++)
        {
            cin>>term;
            sum+=term;
        }
    }
    Series(int a,int b)
    {
        sum=0;
        int term;
        for(int i=a;i<=b;i++)
        {

            sum+=i;
        }
    }
    void display()
    {
        cout<<"\nsum of given series is: "<<sum;
    }
};
int main()
{
    Series s1(2);
    s1.display();
    Series s2(3,5);
    s2.display();
}
```

Input and Output