```
// CPP program to illustrate
// Operator Overloading
#include<iostream>
using namespace std;

class Complex {
private:
    int real, imag;
public:
    Complex(int r = 0, int i =0)  {real = r;   imag = i;}

    // This is automatically called when '+' is used
    // between two Complex objects
    Complex operator - (Complex const obj) {
         Complex res;
         res.real = real - obj.real;
         res.imag = imag - obj.imag;
         return res;
    }


    void print() { cout << real << " + i" << imag << endl; }
};

int main()
{
    Complex c1(10, 5), c2(2, 4);
    Complex c3 = c1 - c2; // An example call to "operator+"
    c3.print();
       return 0;
}
```

```
#include<iostream>
#include<string>
using namespace std;
class stud
{
	public:
		int rollno,prnno;
		string fname,lname,address,branch,mobno;
	stud()
	{
		rollno=0;
		prnno=0;
		fname=" ";
		lname=" ";
		address=" ";
		branch=" ";
		mobno=" ";
	}
	void read()
	{
		cout<<"Enter rollno";
		cin>>rollno;
		cout<<"Enter prn no";
		cin>>prnno;
		cout<<"Enter first name";
		cin>>fname;
		cout<<"Enter last name";
		cin>>lname;
		cout<<"Enter address";
		cin>>address;
```

```
                cout<<"Enter branch";

                cin>>branch;

                cout<<"Enter mob no";

                cin>>mobno;

        }

        stud operator + (stud s)

        {

                stud t;

                t.fname=s.fname+fname;

                t.lname=s.lname+lname;

                return t;

        }

        void display()

        {

                cout<<"\n Rollno"<<rollno;

                cout<<"\n Prn no"<<prnno;

                cout<<"\n First name"<<fname;

                cout<<"\n Last name"<<lname;

                cout<<"\n Address"<<address;

                cout<<"\n Branch"<<branch;

                cout<<"\n Mob no"<<mobno;

        }


};

int main()

{

        stud h,v,b;

        v.read();

        v.display();

        b.read();

        b.display();
```

```
       h=v+b;
       cout<<"\n Concatenated First Name"<<h.fname;
       cout<<"\n Concatenated Last Name"<<h.lname;
return 0;
}
```

```
#include<iostream>
#include<cstring>
using namespace std;
 class String
{
  char str[20]; //member variable for string input
public:
  void input() //member function
  {
    cout<<"Enter your string: ";
    cin.getline(str,20);
  }
  void display() //member function for output
  {
    cout<<"String: "<<str;
  }
 String operator+(String s) //overloading
  {
    String obj;
    strcat(str,s.str);
    strcpy(obj.str,str);
    return obj;
  }
};
```

```
int main()
{
   String str1,str2,str3; //creating three object
   str1.input();
   str2.input();
   str3=str1+str2;
   str3.display(); //displaying
   return 0;
}
```

```
#include<iostream>
#include<cstring>
using namespace std;
class String
{
           char str[100];
           int len;
     public :
           void read();                    // for reading string
           void print();                   // for printing string
// for overloading equal to operator for equality of two string
      int operator == (String);
};
void String::read()
{
   cout << "Enter your string : " ;
   cin >> str;
   len=strlen(str) ;
}
  // Function to print the string
```

```
void String :: print()
{
   cout << "Your string is " << str << endl ;
}
// Definition for equal to operator
int String :: operator == (String s)
{
   if (strcmp(str,s.str)==0)
          return 1 ;
   else
      return 0 ;
}
int  main()
{
   String s1,s2,s3;
   s1.read();
   s2.read();
  if(s1 == s2)   // call operator function to compare //two strings.
          cout << "Both strings are same" <<endl ;
   else
          cout << "Both strings are different" <<endl ;
   return 0;
}
```