1. **Overloading of insertion and extraction operator in c++.**

**#include <iostream>**

**using namespace std;**

**class Complex{**

**int x,y;**

**public:**

**friend ostream& operator<<(ostream &print, Complex c1);**

**friend istream& operator>>(istream &scan, Complex &c1);**

**};**

**ostream& operator<<(ostream &print, Complex c1){**

**print<<" value of x = "<<c1.x<<" and value of y ="<<c1.y;**

**return print;**

**}**

**istream& operator>>(istream &scan, Complex &c1){**

**cout<<"Enter The value of x ";**

**scan>>c1.x;**

**cout<<"Enter The value of y ";**

**scan>>c1.y;**

**return scan;**

**}**

**int main(){**

**Complex c1,c2;**

**cin>>c1>>c2;**

**cout<<c1<<c2;**

**return 0;**

**}**

1. **Overloading of unary operator and +operator.**

**#include <iostream>**

**using namespace std;**

**class complex{**

**int x,y;**

**public:**

**void insert(int x,int y){**

**this->x = x;**

**this->y = y;**

**}**

**void disply(){**

**cout<<"Value of x = "<<x <<" and value of y ="<<y<<endl;**

**}**

**friend complex operator+(int x , complex c);**

**friend complex operator++(complex c);**

**};**

**complex operator+(int x , complex c){**

**complex temp;**

**temp.x = x + c.x;**

**temp.y = x + c.y;**

**return temp;**

**}**

**complex operator++(complex c){**

**complex temp;**

**temp.x =c.x+1;**

**temp.y =c.y+1;**

**return temp;**

**}**

**int main(){**

**complex c1,c2,c3;**

**c1.insert(10,20);**

**c1.disply();**

**c2.insert(20,30);**

**c2.disply();**

**c3 = operator+(18,c2);**

**c3.disply();**

**}**

1. **Program to Demonstrate Copy Constructor Program**

**#include <iostream>**

**using namespace std;**

**class Copy{**

**int x,y;**

**public:**

**Copy(){**

**x = y = 0;**

**}**

**void insert(int x,int y){**

**this->x =x;**

**this->y =y;**

**}**

**void display(){**

**cout<<"Value of x = "<<x <<" and value of y ="<<y<<endl;**

**}**

**Copy(Copy &c){**

**x = c.x;**

**y = c.y;**

**}**

**};**

**int main(){**

**Copy c1,c2,c3;**

**c1.insert(10,20);**

**c2 = c3 = c1;**

**c1.display();**

**c2.display();**

**c3.display();**

**}**

1. **Program to overload + operator for addition of Two matrices.**

**#include <iostream>**

**using namespace std;**

**class matrics{**

**int x[3][3];**

**public:**

**void insert(){**

**int y;**

**for(int i=0;i<3;i++){**

**for(int j=0;j<3;j++){**

**cout<<"Enter value ["<<i<<"]"<<"["<<j<<"]";**

**cin>>y;**

**x[i][j] = y;**

**}**

**cout<<endl;**

**}**

**}**

**void display(){**

**cout<<endl;**

**for(int i=0;i<3;i++){**

**for(int j=0;j<3;j++){**

**cout<<x[i][j]<<" ";**

**}**

**cout<<endl;**

**}**

**}**

**matrics operator+(matrics m1){**

**matrics temp;**

**for(int i=0;i<3;i++){**

**for(int j=0;j<3;j++){**

**temp.x[i][j] = x[i][j] + m1.x[i][j];**

**}**

**}**

**return temp;**

**}**

**};**

**int main(){**

**matrics m1,m2,m3;**

**m1.insert();**

**m1.display();**

**m2.insert();**

**m2.display();**

**m3 = m1+m2;**

**m3.display();**

**}**

**5. program to overload (+,-.\*,/,%,==)operator.**

**#include <iostream>**

**using namespace std;**

**class complex{**

**private:**

**int a,b;**

**public:**

**complex(){ a = b =0;}**

**void setter(int,int);**

**void getter(){**

**cout<<"value of a = "<<a<< " and value of b ="<<b<<endl;**

**}**

**void getValue(){**

**if(a == 0 || b == 0){**

**cout<<"Both Are Unequal"<<endl;**

**}**

**else{**

**cout<<"Both Are Equal";**

**}**

**}**

**complex operator++(int);**

**complex operator+(complex c1);**

**complex operator-(complex c1);**

**complex operator\*(complex c1);**

**complex operator/(complex c1);**

**complex operator%(complex c1);**

**complex operator==(complex c1);**

**};**

**void complex::setter(int a,int b){**

**this->a = a;**

**this->b = b;**

**}**

**complex complex::operator+(complex c1){**

**complex temp;**

**temp.a = a+c1.a;**

**temp.b = b+c1.b;**

**return temp;**

**}**

**complex complex::operator-(complex c1){**

**complex temp;**

**temp.a = a-c1.a;**

**temp.b = b-c1.b;**

**return temp;**

**}**

**complex complex::operator\*(complex c1){**

**complex temp;**

**temp.a = a\*c1.a;**

**temp.b = b\*c1.b;**

**return temp;**

**}**

**complex complex::operator/(complex c1){**

**complex temp;**

**temp.a = a/c1.a;**

**temp.b = b/c1.b;**

**return temp;**

**}**

**complex complex::operator%(complex c1){**

**complex temp;**

**temp.a = a%c1.a;**

**temp.b = b%c1.b;**

**return temp;**

**}**

**complex complex::operator==(complex c1){**

**complex temp;**

**temp.a = a == c1.a;**

**temp.b = b == c1.b;**

**return temp;**

**}**

**complex complex::operator++(int){**

**complex temp;**

**temp.a = a+1;**

**temp.b = b+1;**

**return temp;**

**}**

**int main(){**

**complex c1,c2,c3;**

**c1.setter(10,20);**

**c2.setter(10,20);**

**c1.getter();**

**c2.getter();**

**c3 = c1+c2;**

**c3.getter();**

**c3 = c1 == c2;**

**c3.getValue();**

**c3 = c1++;**

**c3.getter();**

**}**

**6.program to overload ! operator which is used to convert the given character of string into lowercase (if characters are in uppercase) or vice - versa. For eg(“my name is Abhay Raj”)**

**Output (“MY NAME IS aBHAY rAH”)**

**#include <iostream>**

**using namespace std;**

**#include <string.h>**

**class mystr{**

**char str[100];**

**public:**

**void setter(char str1[]){**

**strcpy(str,str1);**

**}**

**void getter(){**

**cout<<" "<<str;**

**cout<<endl;**

**}**

**mystr operator!(){**

**mystr st;**

**int x = strlen(str);**

**for(int i=0;i<=x;i++){**

**if(str[i] >=65 && str[i]<=90){**

**st.str[i] = str[i]+32;**

**}**

**else if(str[i]>=97 && str[i]<=122){**

**st.str[i] = str[i]-32;**

**}**

**else**

**st.str[i] = str[i];**

**}**

**return st;**

**}**

**};**

**int main(){**

**mystr s1,s3;**

**s1.setter("my name is abhay raj");**

**s3 = !s1;**

**s1.getter();**

**s3.getter();**

**}**