**Assignment – 27**

**Operator overloading and friend function**

**1.** **#include<iostream>**

**using namespace std;**

**class Complex**

**{**

**int num1,num2;**

**public:**

**void accept()**

**{**

**cout<<"\nENTER TWO CPMPLEX NUMBERS:";**

**cin>>num1>>num2;**

**}**

**friend Complex operator+(Complex c1,Complex c2);**

**void display()**

**{**

**cout<<num1<<"+"<<num2<<"i"<<"\n";**

**}**

**};**

**Complex operator+(Complex c1,Complex c2)**

**{**

**Complex c;**

**c.num1=c1.num1+c2.num1;**

**c.num2=c1.num2+c2.num2;**

**return(c);**

**}**

**int main()**

**{**

**Complex c1,c2,sum;**

**c1.accept();**

**c1.display();**

**sum=c1+c2;**

**cout<<"\n ENTERED VALUES:\n";**

**return 0;**

**}**

**2.** **#include <iostream>**

**using namespace std;**

**class NUM {**

**private:**

**int n;**

**public:**

**void getNum(int x)**

**{**

**n = x;**

**}**

**void dispNum(void)**

**{**

**cout << "value of n is: " << n;**

**}**

**void operator++(void)**

**{**

**n = ++n;**

**}**

**void operator--(void)**

**{**

**n = --n;**

**}**

**};**

**int main()**

**{**

**NUM num;**

**num.getNum(10);**

**++num;**

**cout << "After increment - ";**

**num.dispNum();**

**cout << endl;**

**--num;**

**cout << "After decrement - ";**

**num.dispNum();**

**cout << endl;**

**return 0;**

**}**

**3.** **#include<iostream>**

**using namespace std;**

**class complex**

**{**

**int real,imag;**

**public:**

**void set()**

**{**

**cout<<"enter real and img part";**

**cin>>real>>imag;**

**}**

**friend complex sum(complex,complex);**

**void display();**

**};**

**void complex::display()**

**{**

**cout<<"the sum of complex num is"<<real<<"+i"<<imag;**

**}**

**complex sum(complex a,complex b)**

**{**

**complex t;**

**t.real=a.real+b.real;**

**t.imag=a.imag+b.imag;**

**return t;**

**}**

**int main()**

**{**

**complex a,b,c;**

**a.set();**

**b.set();**

**c=sum(a,b);**

**c.display();**

**return(0);**

**}**

**4.** **#include<iostream>**

**#include<stdio.h>**

**using namespace std;**

**class Time**

**{**

**private:**

**int hours,minutes,seconds;**

**public:**

**Time()**

**{**

**hours=minutes=seconds=0;**

**}**

**friend int operator>>(istream &input,Time &t)**

**{**

**cout<<"\n Enter hours:";**

**input>>t.hours;**

**cout<<"\n Enter minutes:";**

**input>>t.minutes;**

**cout<<"\n Enter seconds:";**

**input>>t.seconds;**

**t.minutes=t.minutes+t.seconds/60;**

**t.seconds%=60;**

**t.hours=t.hours+t.minutes/60;**

**t.minutes%=60;**

**if(t.hours>=25)**

**return 1;**

**else**

**return 0;**

**}**

**friend void operator<<(ostream &Output, Time &t)**

**{**

**Output<<"\n Hours:"<<t.hours;**

**Output<<"\n Minutes:"<<t.minutes;**

**Output<<"\nSeconds:"<<t.seconds;**

**}**

**int operator==(Time t1)**

**{**

**int tot=hours\*3600+minutes\*60+seconds;**

**int tot1=t1.hours\*3600+t1.minutes\*60+t1.seconds;**

**if(tot=tot1)**

**return 1;**

**else**

**return 0;**

**}**

**~Time(){};**

**};**

**int main()**

**{**

**Time t,t1;**

**cout<<"\nEnter first time";**

**cout<<"\n------------------------";**

**if(cin>>t)**

**{**

**cout<<"\n Invalid Time";**

**return 0;**

**}**

**cout<<"\n first time";**

**cout<<t;**

**cout<<"Enter second time:";**

**cout<<"\n-----------------";**

**if(cin>>t1)**

**{**

**cout<<"\n Invalid Time";**

**return 0;**

**}**

**cout<<"\n Second Time";**

**cout<<t1;**

**if(t==t1)**

**cout<<"\n\nTimes are same";**

**else**

**cout<<"\n\nTimes are Differtent";**

**return 0;**

**}**

**5.** **#include<iostream>**

**using namespace std;**

**class Numbers**

**{**

**int x,y,z;**

**public:**

**void accept()**

**{**

**cout<<"\nEnter three Numbers;";**

**cout<<"\n-------------------------------";**

**cout<<"\nFirst Numbers:";**

**cin>>x;**

**cout<<"\n Second Number:";**

**cin>>y;**

**cout<<"\nThird Number:";**

**cin>>z;**

**cout<<"\n------------------------------";**

**}**

**void display()**

**{**

**cout<<" "<<x<<" "<<y<<" "<<z;**

**}**

**void operator-()**

**{**

**x=-x;**

**y=-y;**

**z=-z;**

**}**

**};**

**int main()**

**{**

**Numbers num;**

**num.accept();**

**cout<<"\n Numbers are:\n\n";**

**num.display();**

**-num;**

**cout<<"\n\n Negated Numbers are:\n\n";**

**num.display();**

**return 0;**

**}**

**6.** **#include<iostream>**

**#include<string.h>**

**using namespace std;**

**class CString**

**{**

**public:**

**char str[20];**

**public:**

**void get\_string()**

**{**

**cout<<"\n Enter String : ";**

**cin>>str;**

**}**

**void display()**

**{**

**cout<<str;**

**}**

**CString operator+(CString x)**

**{**

**CString s;**

**strcat(str, x.str);**

**strcpy(s.str, str);**

**return s;**

**}**

**int operator==(CString &t);**

**};**

**int CString::operator==(CString &t)**

**{**

**for(int i=0; str[i]!='\_'; i++)**

**{**

**for(int j=0; t.str[j]!='\_'; j++)**

**{**

**if(str[i]==t.str[j])**

**{**

**return 0;**

**}**

**else**

**{**

**return 1;**

**}**

**}**

**}**

**}**

**int main()**

**{**

**CString str1, str2, str3;**

**int result=0;**

**str1.get\_string();**

**str2.get\_string();**

**cout<<"\n ----------------------------------------------";**

**cout<<"\n\n First String is : ";**

**str1.display();**

**cout<<"\n\n Second String is : ";**

**str2.display();**

**cout<<"\n ----------------------------------------------";**

**str3=str1+str2;**

**cout<<"\n\n Concatenated String is : ";**

**str3.display();**

**result=str1==str2;**

**if(result==0)**

**{**

**cout<<"\n\n Both Strings are Equal";**

**}**

**else**

**{**

**cout<<"\n\n Both Strings are Not Equal";**

**}**

**return 0;**

**}**

**7.** **#include<iostream>**

**#include<stdio.h>**

**using namespace std;**

**class fraction**

**{**

**private:**

**long num, deno;**

**public:**

**fraction(long int n=0, long int d=0)**

**{**

**num=n;**

**deno=d;**

**}**

**friend void operator>>(istream &in, fraction &f)**

**{**

**cout<<"\n Numerator : ";**

**in>>f.num;**

**cout<<"\n Denominator : ";**

**in>>f.deno;**

**}**

**friend void operator<<(ostream &out, fraction &f)**

**{**

**out<<f.num<<"/"<<f.deno;**

**}**

**fraction operator++()**

**{**

**++num;**

**++deno;**

**return (\*this);**

**}**

**fraction operator++(int s)**

**{**

**fraction tmp=\*this;**

**this->num++;**

**this->deno++;**

**return tmp;**

**}**

**};**

**int main()**

**{**

**fraction f1,f2;**

**cout<<"\n f1 : ";**

**cout<<f1;**

**cout<<"\n f2 : ";**

**cout<<f2;**

**cout<<"\n\n Enter 1st Fraction Value \n";**

**cin>>f1;**

**cout<<"\n f1++ : ";**

**f1++;**

**cout<<f1;**

**<<"\n ++f1 : ";**

**++f1;**

**cout<<f1;**

**cout<<"\n\n Enter 2nd Fraction Value \n";**

**cin>>f2;**

**f2=++f1;**

**cout<<"\n f2 = ++f1";**

**cout<<"\n f1 : ";**

**cout<<f1;**

**cout<<"\n f2 : ";**

**cout<<f2;**

**f2=f1++;**

**cout<<"\n\n f2 = f1++";**

**cout<<"\n f1 : ";**

**cout<<f1;**

**cout<<"\n f2 : ";**

**cout<<f2;**

**return 0;**

**}**

**8.** **#include<iostream>**

**using namespace std;**

**class Matrix**

**{**

**int a[3][3];**

**public:**

**void accept();**

**void display();**

**void operator-();**

**};**

**void Matrix::accept()**

**{**

**cout<<"\n Enter Matrix Element (3 X 3) : \n\n";**

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

**{**

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

**{**

**cout<<" ";**

**cin>>a[i][j];**

**}**

**}**

**}**

**void Matrix::display()**

**{**

**cout<<"\n Matrix is : \n\n";**

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

**{**

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

**{**

**cout<<" ";**

**cout<<a[i][j]<<"\t";**

**}**

**cout<<"\n";**

**}**

**}**

**void Matrix::operator-()**

**{**

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

**{**

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

**{**

**a[i][j]=-a[i][j];**

**}**

**}**

**}**

**int main()**

**{**

**Matrix m;**

**m.accept();**

**m.display();**

**-m;**

**m.display();**

**return 0;**

**}**

**9.** **#include<iostream>**

**#include<string.h>**

**using namespace std;**

**class mystring**

**{**

**char str[500];**

**public:**

**void operator!();**

**void accept\_string()**

**{**

**cout<<"\n Enter String : ";**

**cin>>str;**

**}**

**void display\_string()**

**{**

**cout<<str;**

**}**

**};**

**void mystring::operator!()**

**{**

**for(int i=0; str[i]!='\_'; i++)**

**{**

**if(str[i]>=65&&str[i]<=96)**

**{**

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

**}**

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

**{**

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

**}**

**}**

**cout<<"\n\n Reverse Case String is : "<<str;**

**}**

**int main()**

**{**

**mystring s1;**

**s1.accept\_string();**

**cout<<"\n\n String is : ";**

**s1.display\_string();**

**!s1;**

**return 0;**

**}**

**10.** **#include<iostream>**

**using namespace std;**

**class Matrix**

**{**

**int a[3][3];**

**public:**

**void accept();**

**void display();**

**void operator +(Matrix x);**

**};**

**void Matrix::accept()**

**{**

**cout<<"\n Enter Matrix Element (3 X 3) : \n";**

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

**{**

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

**{**

**cout<<" ";**

**cin>>a[i][j];**

**}**

**}**

**}**

**void Matrix::display()**

**{**

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

**{**

**cout<<" ";**

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

**{**

**cout<<a[i][j]<<"\t";**

**}**

**cout<<"\n";**

**}**

**}**

**void Matrix::operator +(Matrix x)**

**{**

**int mat[3][3];**

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

**{**

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

**{**

**mat[i][j]=a[i][j]+x.a[i][j];**

**}**

**}**

**cout<<"\n Addition of Matrix : \n\n";**

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

**{**

**cout<<" ";**

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

**{**

**cout<<mat[i][j]<<"\t";**

**}**

**cout<<"\n";**

**}**

**}**

**int main()**

**{**

**Matrix m,n;**

**m.accept();**

**n.accept();**

**cout<<"\n First Matrix : \n\n";**

**m.display();**

**cout<<"\n Second Matrix : \n\n";**

**n.display();**

**m+n;**

**return 0;**

**}**