**NAME: CHANDRIMA BHATTACHARYA**

**EXAM ROLL NO: 111308009**

**CLASS ROLL NO: HX-05**

**DEPARTMENT: I.T.**

**SEMESTER: 5**

**DATE OF ASSIGNMENT: 28/08/2015 & 04/09/2015**

**ASSIGNMENT: 3**

**3.1.1. Check whether a number is even or odd by overloading ! operator.**

**Source Code:**

#include<iostream>

using namespace std;

class A

{

int x;

public:

A()

{

cout<<"Enter a number:";

cin>>x;

}

int operator!();

};

int A::operator!()

{

return(x%2);

}

int main()

{

A a1;

if ( !a1 )

cout << "Is odd\n";

else

cout << " Is even\n";

return 0;

}

**Output:**

Enter a number.

>>7

Is odd

**3.1.2.** **Check whether a number is prime or not by overloading -- operator**

**Source code:**

#include<iostream>

using namespace std;

class A

{

int x;

public:

A()

{

cout<<"Enter a number:";

cin>>x;

}

int operator--();

} ;

int A::operator--()

{

cout<<"x = "<<x<<endl;

int i,c=0;

for(i=2;i<=x/2;i++)

{

//cout<<"i "<<i<<"c "<<c<<endl;

if(x%i==0){

c++;

//cout<<"i "<<i<<" x "<<x<<endl;

}

}

if(c==0)

return 0;

return 1;

}

int main()

{

A a1;

if ( --a1 )

cout << "is not prime\n";

else

cout << " is prime\n";

return 0;

}

**Output:**

Enter a number

>>7

Is a prime

**3.1.3. Add two complex number by overloading + operator a) Using Member function. b) Using Friend Function.**

**Source Code (a):**

#include <iostream>

#include<cstring>

using namespace std;

class comp

{

int i;

int r;

public:

void getdata(int m,int n)

{

r=m;

i=n;

}

void showdata()

{

char c;

if(i<0)

c='-';

else

c='+';

cout<<"\n Complex data: "<<r<<c<<i<<"i";

}

comp operator+(comp c1)

{

comp complx;

complx.r=r+c1.r;

complx.i=c1.i+i;

return complx;

}

};

int main()

{

int im,re;

comp c1,c2,c3;

cout<<"\n Enter real and imaginary part!";

cin>>re>>im;

c1.getdata(re,im);

cout<<"\n Enter real and imaginary part!";

cin>>re>>im;

c1.showdata();

c2.getdata(re,im);

c2.showdata();

c3=c1+c2;

c3.showdata();

return 0;

}

**Output:**

Enter real and complex part!

>>21

>>4

Enter real and complex part!

>>3

>>7

Complex data: 24+11i

**Source code (b):**

#include <iostream>

#include<cstring>

using namespace std;

class comp

{

int i;

int r;

public:

void getdata(int m,int n)

{

r=m;

i=n;

}

void showdata()

{

char c;

if(i>0)

c='+';

cout<<"\n Complex data: "<<r<<c<<i<<"i";

}

friend comp operator+(comp c1,comp c2);

};

comp operator+(comp c1,comp c2)

{

comp complx;

complx.r=c1.r+c2.r;

complx.i=c1.i+c2.i;

return complx;

}

int main()

{

int im,re;

comp c1,c2,c3;

cout<<"\n Enter real and imaginary part!";

cin>>re>>im;

c1.getdata(re,im);

cout<<"\n Enter real and imaginary part!";

cin>>re>>im;

c1.showdata();

c2.getdata(re,im);

c2.showdata();

c3=c1+c2;

c3.showdata();

return 0;

}

**Output:**

Enter real and complex part!

>>21

>>4

Enter real and complex part!

>>3

>>-7

Complex data: 24-3i

**Revision Problem**

**1.** **Class Distance consists of length in feet and inches. Class Distance contains one default constructor one parameterized constructor function getdata() to take the value of feet and inches. function show() to display. Overload < operator to compare two distances.**

**Source code:**

#include <iostream>

#include <cstring>

using namespace std;

class distance1

{

int feet;

int inch;

public:

distance1()

{

feet=0;

inch=0;

}

distance1(int m)

{

feet=m;

inch=m;

}

void getdata(int m,int n)

{

feet=m;

inch=n;

}

void show()

{

cout<<"\n distance is= "<<feet<<" feet and "<<inch<<" inches. ";

}

distance1 operator<(distance1 b)

{

if(inch<12 and b.inch<12)

if(feet>b.feet)

cout<<"\n distance entered first is larger";

else

cout<<"\n distance two is larger";

else if(feet=b.feet)

{

if(inch>b.inch)

cout<<"\n distance entered first is larger";

else

cout<<"\n distance two is larger";

}

}

};

int main()

{

distance1 a,b,c;

int ft,in;

cout<<"\n Enter distance 1 where feet=inches";

cin>>ft;

a.getdata(ft,ft);

cout<<"\n Enter distance 2 as feet and inches";

cin>>ft>>in;

b.getdata(ft,in);

a.show();

b.show();

c=a<b;

return 0;

}

**Output:**

Enter distance 1 where feet=inches

>>4

Enter distance 2 as feet and inches

>>7

>>2

Distance is 4 feet 4 inches.

Distance is 7 feet 2 inches.

Distance 2 is greater.

**2. Overload += operator in the Distance class.**

**Source code:**

#include <iostream>

#include <cstring>

using namespace std;

class distance1

{

int feet;

int inch;

public:

distance1()

{

feet=0;

inch=0;

}

distance1(int m)

{

feet=m;

inch=m;

}

void getdata(int m,int n)

{

feet=m;

inch=n;

}

void show()

{

if(inch>=12)

{

feet++;

inch-=12;

}

cout<<"\n distance is= "<<feet<<" feet and "<<inch<<" inches. ";

}

distance1 operator+=(distance1 b)

{

distance1 dis;

dis.inch=inch+b.inch;

dis.feet=feet+b.feet;

if(dis.inch>=12)

{

dis.feet+=1;

dis.inch-=12;

}

return dis;

}

};

int main()

{

distance1 a,b,c;

int ft,in;

cout<<"\n Enter distance 1 where feet=inches";

cin>>ft;

a.getdata(ft,ft);

cout<<"\n Enter distance 2 as feet and inches";

cin>>ft>>in;

b.getdata(ft,in);

a.show();

b.show();

c=a+=b;

c.show();

return 0;

}

**Output:**

Enter distance 1 where feet=inches

>>4

Enter distance 2 as feet and inches

>>7

>>21

Distance is 4 feet 4 inches.

Distance is 7 feet 2 inches.

Distance is 11 feet 6 inches.

**3.** **Concatenate two strings by overloading + operator.**

**Source code:**

#include <iostream>

#include <cstring>

#include<cstdlib>

using namespace std;

class concstr

{

char a[50];

public:

void getdata()

{

cout<<"\n Enter string";

cin>>a;

}

void showdata()

{

cout<<"\n the string is "<<a;

}

concstr operator+(concstr b)

{

concstr s;

strcat(a,b.a);

strcpy(s.a,a);

return s;

}

};

int main()

{

concstr A,b,c;

A.getdata();

b.getdata();

A.showdata();

b.showdata();

c=A+b;

c.showdata();

return 0;

}

**Output:**

Enter string.

>>Hi!

Enter string.

>>How are you doing?

The string is: Hi!

The string is: How are you doing?

The string is: Hi! How are you doing?

**4.** **. Overload ++ as prefix (++c1) and postfix (c1++) in some class**

**Source code:**

#include <iostream>

using namespace std;

class fixes

{

int c;

public:

fixes(){}

fixes(int n){

c=n;

}

fixes operator++(){

c++;

return c;

}

fixes operator++(int){

fixes a(c);

(c)++;

return c;

}

void showdata()

{

cout<<"\n No= "<<c;

}

};

int main()

{

int n;

cout<<"\n Enter num";

cin>>n;

fixes a(n);

a++;

a.showdata();

++a;

a.showdata();

return 0;

}

**Output:**

Enter no >> 4

No=5

No=6

**5.** **Overload == operator to compare two strings.**

**Source code:**

#include <iostream>

#include <cstring>

#include<cstdlib>

using namespace std;

class concstr

{

char a[50];

public:

void getdata(){

cout<<"\n Enter string";

cin>>a;

}

void showdata()

{

cout<<"\n the string is "<<a;

}

concstr operator==(concstr b)

{

concstr s;

if(strcmp(a,b.a)==0)

cout<<"\n The strings are same!";

else

cout<<"\n String is not same!";

}

};

int main()

{

concstr A,b,c;

A.getdata();

b.getdata();

A.showdata();

b.showdata();

c=A==b;

return 0;

}

**Output:**

Enter string >>Hi!

Enter string >>hI!

String is not same.

**ASSIGNMENT 3 (CONT…)**

**3.2.1.** **Write a program to convert a distance entered in Feet and Inches to Meter using class to basic data type conversion.**

**Source Code:**

#include <iostream>

using namespace std;

class dist

{

private:

int feet;

int inch;

public:

dist()

{}

void getdata()

{

cout<<"\n Enter feet and inch";

cin>>feet>>inch;

}

operator float()

{

float m;

m=(.3048\*feet)+(.0254\*inch);

return m;

}

};

int main()

{

float result;

dist d1;

d1.getdata();

result=d1;

cout<<"\n Basic conversation ="<<result<<”metres”;

return 0;

}

**Output:**

Enter feet and inches

>>21 >>4

Basic conversion =6.5024 metres

**3.2.2. Two classes one is Civil\_Time and Another is Railway\_Time. Enter hours and minutes in Railway time(24 hour format) and display the time in Civil time(12 hour format with a.m. and p.m.) using one class type to another class type conversion.**

**Source Code:**

#include <iostream>

#include <cstring>

using namespace std;

class central\_time;

class railway\_time

{

int hour;

int min;

public:

railway\_time()

{

cout<<"Enter time as hour and min in 24 hr format";

cin>>hour>>min;

}

int gethour()

{

return hour;

}

int getmin()

{

return min;

}

};

class central\_time

{

int h,m,flag;

public:

central\_time(railway\_time x)

{

if(x.gethour()>=12)

{

flag=1;

h=x.gethour()-12;

}

else if (x.hourhour()==0)

{

flag=0;

h=12;

}

m=x.getmin();

}

void show()

{

if(flag==1)

cout<<"\n Time is "<<h<<":"<<m<<"pm";

else

cout<<"\n Time is "<<h<<":"<<m<<"am";

}

};

int main()

{

railway\_time c1;

central\_time x=c1;

x.show();

return 0;

}

**Output:**

Enter time as hour and min in 24 hr format

>>0

>>00

Time is 0.00 a.m.