Assignment 29

1.Write a C++ program to convert Primitive type to Complex type.

Example -

int main()

{

Complex c1;

Int x=5;

c1=x;

return 0;

}

#include<iostream>

using namespace std;

class Complex

{

    int real;

    int img;

    public:

    Complex()

    {

    }

    Complex(int x)

    {

        real=x;

        img=x;

    }

    void display()

    {

        cout<<"Real = "<<real<<"\n Img = "<<img<<endl;

    }

};

int main()

{

    Complex c1;

    int x=5;

    c1=x;//parameterized constructor is called

    c1.display();

    return 0;

}

Output:

t\_29\" ; if ($?) { g++ type\_casting\_complex.cpp -o type\_casting\_complex } ; if ($?) { .\type\_casting\_complex }

Real = 5

Img = 5

PS C:\Users\tusha\Documents\coadind\assigmen

2. Write a C++ program to convert Complex type to Primitive type.

Example -

int main()

{

Complex c1;

c1.setData(3,4);

int x;

x=c1;

return 0;

}

#include<iostream>

using namespace std;

class Complex

{

    int real;

    int img;

    public:

        Complex()

        {

        }

        Complex(int x, int y)

        {

            real =x;

            img=y;

        }

        void setData(int x,int y)

        {

            real=x;

            img=y;

        }

explicitly you can do like this

explicit operator int(){

return real;

}

--------------

int main()

{

x=int(c1);

}

        operator int()

        {

            return real;

        }

};

int main()

{

    Complex c1;

    c1.setData(3,4);

    int x;

    x=c1;// implicite type casting (explicitly -> x= (int)c1) this is also valid

    cout<<"x value is  :  "<<x<<endl;

    return 0;

}

Ouput:

"c:\Users\tusha\Documents\coadind\assigment\_29\" ; if ($?) { g++ compelx\_int.cpp -o compelx\_int } ; if ($?) { .\compelx\_int }

x value is : 3

PS C:\Users\tusha\Documents\coadind\assigment\_29>

3. Create a Product class and convert Product type to Item type using constructor

int main()

{

Item i1;

Product p1;

p1.setData(3,4);

i1=p1;

return 0;

}

 #include<iostream>

 using namespace std;

class Product

 {

    int x;

    int y;

    public:

        Product()

        {

        }

        Product(int a,int b)

        {

            x=a;

            y=b;

        }

        int get\_x()

        {

            return x;

        }

        int get\_y()

        {

            return y;

        }

        void display()

        {

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

        }

 };

 class Item

 {

    int  a;

    public:

        Item()

        {

        }

        void display()

        {

            cout<<"a  :  "<<a<<endl;

        }

        Item(Product p)

        {

            a=(p.get\_x()+ p.get\_y());

        }

 };

int main()

{

    Product p1(3,4);

    Item i1;

    i1=p1;

    i1.display();

    p1.display();

    return 0;

}

Output:

cd "c:\Users\tusha\Documents\coadind\assigment\_29\" ; if ($?) { g++ item\_to\_product.cpp -o item\_to\_product } ; if ($?) { .\item\_to\_product }

a : 7

x = 3 y = 4

PS C:\Users\tusha\Documents\coadind\assigment\_29>

4. Create Product class and convert Product type to Item type using casting operator

int main()

{

Item i1;

Product p1;

p1.setData(3,4);

i1=p1;

return 0;

}

 #include<iostream>

 using namespace std;

 class Item

 {

    int a;

    public:

        Item()

        {

        }

        Item(int x)

        {

            a=x;

        }

        void display()

        {

            cout<<" a : "<<a<<endl;

        }

 };

 class Product

 {

    int x;

    int y;

    public:

        Product()

        {

        }

        Product(int a, int b)

        {

            x=a;

            y=b;

        }

        void set(int a, int b)

        {

            x=a;

            y=b;

        }

        void display()

        {

            cout<<"x  :  "<<x<<"  y  ;  "<<y<<endl;

        }

        operator Item()

        {

            Item i(x+y);

            return i;

        }

 };

int main()

{

    Item i1;

    Product p1;

    p1.set(3,4);

    i1=p1;

    i1.display();

    p1.display();

    return 0;

}

Output:

d "c:\Users\tusha\Documents\coadind\assigment\_29\" ; if ($?) { g++ item\_operator\_operator.cpp -o item\_operator\_operator } ; if ($?) { .\item\_operator\_operator }

a : 7

x : 3 y ; 4

PS C:\Users\tusha\Documents\coadind\assigment\_29>

5. Create two classes Invent1 and Invent2 and also add necessary constructors in it. Now add

functions to support Invent1 to float and Invent1 to Invent2 type.

Example -

int main()

{

Invent1 s1=(4,5);

Invent2 d1;

float tv;

tv=s1;

d1=s1;

return 0;

}

 #include<iostream>

 using namespace std;

 class Invent1

 {

    int x;

    int y;

    public:

    Invent1()

    {

    }

    Invent1(int a, int b)

    {

        x=a;

        y=b;

    }

    int get\_x()

    {

        return x;

    }

    int get\_y()

    {

        return y;

    }

    void display()

    {

        cout<<" x  : "<<x<<" , y  : "<<y<<endl;

    }

    operator float()

    {

        return(x+y);

    }

 };

 class Invent2

 {

    int x;

    int y;

    public:

        Invent2()

        {

        }

        Invent2(int a, int b)

        {

            x=a;

            y=b;

        }

        void display()

        {

            cout<<" x  : "<<x<<" , y  : "<<y<<endl;

        }

        Invent2( Invent1 s)

        {

            x=s.get\_x();

            y=s.get\_y();

        }

 };

int main()

{

Invent1 s1(4,5);

Invent2 d1;

float tv;

tv=s1;

d1=s1;

cout<<tv<<endl;

s1.display();

d1.display();

return 0;

}

Output:

d "c:\Users\tusha\Documents\coadind\assigment\_29\" ; if ($?) { g++ ivert\_to\_float.cpp -o ivert\_to\_float } ; if ($?) { .\ivert\_to\_float }

9

x : 4 , y : 5

x : 4 , y : 5

PS C:\Users\tusha\Documents\coadind\assigment\_29>

6. Create a Time class and take Duration in seconds. Now you need to convert seconds(i.e in

int ) to Time class.

Example-

int main()

{

int duration;

cout<<”Enter time duration in minutes”;

cin>>duration;

Time t1 = duration;

t1.display();

return 0;

}

 #include<iostream>

 using namespace std;

 class Time

 {

    int day;

    int hr;

    int sec;

    int min;

    public:

        Time()

        {

        }

        Time(int x)

        {

            hr=x/60;

            min = x%60;

            day=hr/24;

            hr=hr%24;

        }

        void display()

        {

            cout<<"Days  : "<<day<<endl;

            cout<<"Hours : "<<hr<<endl;

            cout<<"Min   : "<<min<<endl;

        }

 };

int main()

{

int duration;

cout<<"enter the duration in minutes : "<<endl;

cin>>duration;

Time t1 = duration;

t1.display();

return 0;

}

Output;

enter the duration in minutes :

8000

Days : 5

Hours : 13

Min : 20

PS C:\Users\tusha\Documents\coadind\assigment\_29>

8. Create a Rupee class and convert it into int. And Display it.

Example-

int main()

{

Rupee r = 10;

int x = r;

cout<<x;

return 0;

}

#include<iostream>

using namespace std;

class Rupee

{

    int x;

    public:

        Rupee()

        {

            //default constructor

        }

        Rupee(int n)

        {

            x=n;

        }

        void display()

        {

            cout<<x<<endl;

        }

        operator int()

        {

            return x;

        }

};

int main()

{

Rupee r = 10;

int x = r;//implicitly calling int()

cout<<x;

return 0;

}

Output:

c:\Users\tusha\Documents\coadind\assigment\_29\" ; if ($?) { g++ ruppe\_to\_int.cpp -o ruppe\_to\_int } ; if ($?) { .\ruppe\_to\_int }

10

PS C:\Users\tusha\Documents\coadind\assigment\_29>

9. Create a Dollar class and add necessary functions to support int to Dollar type conversion.

Example-

int main()

{

int x = 50;

Dollar d;

d = x;

d.display();

return 0;

}

#include<iostream>

using namespace std;

class Dollar

{

    int x;

    public:

        Dollar()

        {

            // default consructor

        }

        Dollar(int n)

        {

            x=n;

        }

        void display()

        {

            cout<<"Dollar  :  "<<x<<endl;

        }

};

int main()

{

    int x = 50;

    Dollar d;

    d = x;// this can be done by parameterised constructor

    d.display();

    return 0;

 }

Output:

c:\Users\tusha\Documents\coadind\assigment\_29\" ; if ($?) { g++ int\_to\_dollar.cpp -o int\_to\_dollar } ; if ($?) { .\int\_to\_dollar }

Dollar : 50

PS C:\Users\tusha\Documents\coadind\assigment\_29>

10. Create two classes Rupee and Dollar and add necessary functions to support Rupee to

Dollar and Dollar to Rupee conversion.

Example-

int main()

{

Rupee r = 23;

Dollar d = r; // Rupee to Dollar conversion

d.display();

r.display();

r = d; // Dollar to Rupee Conversion

d.display();

r.display();

return 0;

}

#include<iostream>

using namespace std;

class Rupee

{

    float x;

    public:

        Rupee()

        {

            //DC called

        }

        Rupee(float n)

        {

            x=n;

        }

        float get\_R()

        {

            return x;

        }

        void display()

        {

            cout<<" Rupee  :  "<<x<<endl;

        }

        operator float()

        {

            return x;//assuming 1 dollar equal to 100 INR

        }

};

class Dollar

{

    float x;

    public:

        Dollar()

        {

            //DC called

        }

        Dollar(float n)

        {

            x=n;

        }

        Dollar(Rupee r)

        {

            x=r.get\_R()/100;

        }

        operator Rupee()

        {

            return x\*100;

        }

        void display()

        {

            cout<<"Dollar  :  "<<x<<endl;

        }

        operator float()

        {

            return x;

        }

};

int main()

{

    Rupee r = 23;

    Dollar d = r; // Rupee to Dollar conversion

    d.display();

    r.display();

    r = d; // Dollar to Rupee Conversion

    d.display();

    r.display();

    return 0;

}

Output:

"c:\Users\tusha\Documents\coadind\assigment\_29\" ; if ($?) { g++ ruppe\_to\_dollar.cpp -o ruppe\_to\_dollar } ; if ($?) { .\ruppe\_to\_dollar }

Dollar : 0.23

Rupee : 23

Dollar : 0.23

Rupee : 23

PS C:\Users\tusha\Documents\coadind\assigment\_29>