**Assignment-29 Solution Name: Om Pant**

**Type Casting and Conversion**

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

int main()

{

Complex c1;

Int x=5;

c1=x;

return 0;

}

Ans-

// 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{

    public:

        int x;

        Complex(){

        }

        Complex(int x){

            this->x = x;

        }

        void display(){

            cout<<x;

        }

};

int main(){

    Complex c1;

    int x=5;

    c1=x;

    c1.display();

    return 0;

}

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;

}

Ans-

// 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{

    public:

        int real,img;

        void setData(int x, int y){

            real = x;

            img = y;

        }

        operator int(){

            return real+img;

        }

};

int main(){

        Complex c1;

        c1.setData(3,4);

        int x;

        x=c1;

        cout<<x<<endl;

        return 0;

}

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;

}

Ans-

#include<iostream>

using namespace std;

class Product{

    public:

    string p\_Name;

    int pId;

    void setData(string s, int id){

        p\_Name = s;

        pId = id;

    }

    void display(){

        cout<<"Product Name: "<<p\_Name<<endl<<"Product ID: "<<pId<<endl;

    }

};

class Item{

    public:

    string i\_Name;

    int iId;

        Item(){

        }

        Item(Product p){

            cout<<"Converting Product to Item"<<endl;

            i\_Name = p.p\_Name;

            iId = p.pId;

        }

        void display(){

        cout<<"Item Name: "<<i\_Name<<endl<<"Item ID: "<<iId<<endl;

    }

};

int main()

{

 Item i1;

 Product p1;

 p1.setData("Headphone",0125);

 p1.display();

 i1 = p1;

 i1.display();

 return 0;

}

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;

}

Ans-

#include<iostream>

using namespace std;

class Item{

    public:

    string i\_Name;

    int iId;

        Item(){

        }

        void display(){

        cout<<"Item Name: "<<i\_Name<<endl<<"Item ID: "<<iId<<endl;

    }

};

class Product{

    public:

    string p\_Name;

    int pId;

    void setData(string s, int id){

        p\_Name = s;

        pId = id;

    }

    void display(){

        cout<<"Product Name: "<<p\_Name<<endl<<"Product ID: "<<pId<<endl;

    }

    operator Item(){

        cout<<"Converting Product to Item using Item() Operator"<<endl;

        Item x;

        x.i\_Name = p\_Name;

        x.iId = pId;

        return x;

    }

};

int main()

{

    Item i1;

    Product p1;

    p1.setData("Headphone",0125);

    p1.display();

    i1 = p1;

    i1.display();

    return 0;

}

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 x(4,5);

Invent2 y;

float z;

z = x; // Invent1 to float

y = x; // Invent1 to Invent2

return 0;

}

Ans-

#include<iostream>

using namespace std;

class Invent1{

    public:

    int x,y;

        Invent1(){}

        Invent1(int x, int y){

            this->x = x;

            this->y = y;

        }

        operator float(){

            return x + y;

        }

        void display(){

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

        }

};

class Invent2{

    public:

        int p,q;

        Invent2(){}

        Invent2(Invent1 i){

          p = i.x;

          q = i.y;

        }

        void display(){

            cout<<"p: "<<p<<"p: "<<q<<endl;

        }

};

int main()

{

    Invent1 x(4,5);

    Invent2 y;

    float z;

    z = x; // Invent1 to float

    y = x; // Invent1 to Invent2

    x.display();

    y.display();

    cout<<"z: "<<z<<endl;

    return 0;

}

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

Example

int main()

{

int duration;

cout<<”Enter time duration in minutes”;

cin>>duration;

Time t1 = duration;

t1.display();

return 0;

}

Ans –

#include<iostream>

using namespace std;

class Time{

    int hr = 00, mt = 00, sc = 00;

    public:

        Time(){

        }

        Time(int x){

            if(x>60){

                sc = x%60;

                mt = x/60;

                if(mt>60){

                    hr = mt/60;

                    mt = mt%60;

                }

            }

            else{

                sc = x;

            }

        }

        void showTime(){

            cout<<"Time is: "<<hr<<" hr:"<<mt<<" mt:"<<sc<<" sec "<<endl;

        }

};

int main()

{

    int duration;

    cout<<"Enter time duration in seconds: ";

    cin>>duration;

    Time t1 = duration;

    t1.showTime();

    return 0;

}

7. Create two class Time and Minute and add required getter and setter including constructors. Now you need to type cast Time object into Minute to fetch the minute from Time and display it.

Example -

int main()

{

Time t1(2,30);

t1.display();

Minute m1;

m1.display();

m1=t1 // Fetch minute from time

t1.display();

m1.display();

return 0;

}

Ans-

#include<iostream>

using namespace std;

class Time{

    int hr,mt;

    public:

        int getHour(){

            return hr;

        }

        int getMinute(){

            return mt;

        }

        Time(int h, int m){

            hr = h;

            mt = m;

        }

        void display(){

            cout<<"Time: "<<hr<<" hr "<<mt<<" mt"<<endl;

        }

};

class Minute{

    int mt = 0;

    public:

        Minute(){

        }

        Minute(Time t){

            mt = t.getMinute();

        }

        void display(){

            cout<<"Minute: "<<mt<<" minutes "<<endl;

        }

};

int main()

{

    Time t1(2,30);

    t1.display();

    Minute m1;

    m1.display();

    m1 = t1 ;// Fetch minute from time

    t1.display();

    m1.display();

    return 0;

}

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;

}

Ans-

#include<iostream>

using namespace std;

class Rupee{

    int r;

        public:

        Rupee(int x){

            r = x;

        }

        operator int(){

            return r;

        }

        void display(){

            cout<<"Rupee: "<<r<<endl;

        }

};

int main()

{

    Rupee r = 10;

    int x = r;

    r.display();

    cout<<"x: "<<x<<endl;

    return 0;

}

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;

}

Ans-

#include<iostream>

using namespace std;

class Dollar{

    int x;

    public:

        Dollar(){}

        Dollar(int x){

            this->x = x;

        }

        void display(){

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

        }

};

int main()

{

    int x = 50;

    Dollar d;

    d = x;

    d.display();

    return 0;

}

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;

}

Ans –

#include<iostream>

using namespace std;

class Dollar{

    int d;

    public:

        Dollar(){}

        Dollar(int x){

            d =x;

        }

        int getDollar(){

            return d;

        }

        void display(){

            cout<<"Dollar: "<<d<<endl;

        }

};

class Rupee{

    int ru;

    public:

        int getRupee(){

            return ru;

        }

        Rupee(){

        }

        Rupee(int x){

            ru = x;

        }

        Rupee(Dollar x){ //Dollar to rupee conversion

            ru = x.getDollar();

        }

        //For Rupee to dollar conversion

        operator Dollar(){

            Dollar a =  ru;

            return a;

        }

        void display(){

            cout<<"Rupee: "<<ru<<endl;

        }

};

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;

}