**Assignment – 29 A Job Ready Bootcamp in C++, DSA and IOT MySirG**

**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;

}

Sol – 1.

#include<iostream>

using namespace std;

class Complex

{

private :

int r,i;

public :

Complex(){}

Complex(int x)

{

r=x;

i=x;

}

Complex(int real,int imag)

{

r=real;

i=imag;

}

void printnum()

{

cout<<"Real : "<<r<<" Imag : "<<i<<endl;

}

};

int main()

{

Complex c1;

int x=5;

c1=x;

c1.printnum();

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;

}

Sol – 2.

#include<iostream>

using namespace std;

class Complex

{

private :

int r,i;

public :

Complex(){}

Complex(int x)

{

r=x;

i=x;

}

Complex(int real,int imag)

{

r=real;

i=imag;

}

void printnum()

{

cout<<"Real : "<<r<<" Imag : "<<i<<endl;

}

operator int()

{

return r;

}

};

int main()

{

Complex c1,c2(7,10);

int x=5;

c1=x;

c1.printnum();

x=(int)c2;

cout<<x;

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;

}

Sol – 3.

#include<iostream>

using namespace std;

class Product

{

private :

int a,b;

public :

Product(){}

int geta()

{

return a;

}

void setdata(int x,int y)

{

a=x;

b=y;

}

void print()

{

cout<<"A : "<<a<<" B : "<<b<<endl;

}

};

class Item

{

private :

int c;

public :

Item(){}

Item(Product P)

{

c=P.geta();

}

void print()

{

cout<<"C : "<<c<<endl;

}

};

int main()

{

Item i1;

Product p1;

p1.setdata(3,4);

i1=p1;

i1.print();

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;

}

Sol – 4.

#include<iostream>

using namespace std;

class Product

{

private :

int a,b;

public :

Product(){}

int geta()

{

return a;

}

void setdata(int x,int y)

{

a=x;

b=y;

}

void print()

{

cout<<"A : "<<a<<" B : "<<b<<endl;

}

};

class Item

{

private :

int c;

public :

Item(){}

Item(int x)

{

c=x;

}

void print()

{

cout<<"C : "<<c<<endl;

}

operator Product()

{

Product P;

P.setdata(c,c);

return P;

}

};

int main()

{

Item i1(5);

Product p1;

i1.print();

p1=i1;

p1.print();

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 s1(4,5);

Invent2 d1;

float tv;

tv=s1;

d1=s1;

return 0;

}

Sol – 5.

#include<iostream>

using namespace std;

class Invent1

{

private :

int a,b;

public :

Invent1(){}

Invent1(int x,int y)

{

a=x;

b=y;

}

int geta()

{

return a;

}

int getb()

{

return b;

}

void setdata(int x,int y)

{

a=x;

b=y;

}

void print()

{

cout<<"A : "<<a<<" B : "<<b<<endl;

}

operator float()

{

return a+b;

}

};

class Invent2

{

private :

int c;

public :

Invent2(){}

Invent2(Invent1 I)

{

c=I.geta()+I.getb();

}

Invent2(int x)

{

c=x;

}

void print()

{

cout<<"C : "<<c<<endl;

}

};

int main()

{

Invent1 s1(4,5);

Invent2 d1;

float tv;

tv=s1;

cout<<tv<<endl;

d1=s1;

d1.print();

return 0;

}

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;

}

Sol – 6.

#include<iostream>

using namespace std;

class Time

{

private :

int a=0,b=0,c=0;

public :

Time(int x)

{

a=x/3600;

x%=3600;

b=x/60;

c=x%60;

}

void display()

{

cout<<"Hr : "<<a<<" Min : "<<b<<" Sec : "<<c<<endl;

}

};

int main()

{

int duration;

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

cin>>duration;

Time t1 = duration;

t1.display();

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;

}

Sol – 7.

#include<iostream>

using namespace std;

class Minute

{

private :

int min=0;

public :

Minute(){}

Minute(int a)

{

min=a;

}

void display()

{

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

}

};

class Time

{

private :

int hr,mn;

public :

Time(int hour,int min)

{

hr=hour;

mn=min;

}

void display()

{

cout<<"Hr : "<<hr<<" Min : "<<mn<<endl;

}

operator Minute()

{

Minute M(mn);

return M;

}

};

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;

}

Sol – 8.

#include<iostream>

using namespace std;

class Rupee

{

private :

int x;

public :

Rupee(){}

Rupee(int a)

{

x=a;

}

operator int()

{

return x;

}

};

int main()

{

Rupee r = 10;

int x = r;

cout<<x;

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;

}

Sol – 9.

#include<iostream>

using namespace std;

class Dollar

{

private :

int x;

public :

Dollar(){}

Dollar(int a)

{

x=a;

}

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;

}

Sol – 10.

#include<iostream>

using namespace std;

class Rupee

{

private :

float x;

public :

Rupee(){}

Rupee(float a)

{

x=a;

}

float getr()

{

return x;

}

void display()

{

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

}

operator float()

{

return x;

}

};

class Dollar

{

private :

float x;

public :

Dollar(){}

Dollar(float a)

{

x=a;

}

Dollar(Rupee R)

{

x=R.getr()/80.0;

}

void display()

{

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

}

operator Rupee()

{

Rupee R(x\*80);

return R;

}

};

int main()

{

Rupee r = 80;

Dollar d = r; // Rupee to Dollar conversion

d.display();

r.display();

r = d; // Dollar to Rupee Conversion

d.display();

r.display();

return 0;

}