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.

#include <iostream>

using namespace std;

class Complex

{

int real, img;

public:

Complex() {}

Complex(int x)

{

real = x;

img = x;

}

void display()

{

cout << "Img: " << img << " and Real: " << real << endl;

}

};

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.

#include <iostream>

using namespace std;

class Complex

{

int real, img;

public:

void setData(int x, int y)

{

real = x;

img = y;

}

operator long()

{

return real + img;

}

explicit operator int()

{

return real + img; // you can also return img

}

void display()

{

cout << "Img: " << img << " and Real: " << real << endl;

}

};

int main()

{

Complex c1;

c1.setData(5, 4);

// int x;

// x = c1;

// cout << "Value is: " << x << endl;

// c1.display();

cout << c1 << endl;

return 0;

}

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

int main

#include <iostream>

using namespace std;

class Product

{

int x, y;

public:

void setData(int a, int b)

{

x = a;

y = b;

}

int getX() { return x; }

int getY() { return y; }

void display() { cout << "x = " << x << " y = " << y << endl; }

};

class Item

{

int a, b;

public:

Item() {}

Item(Product p)

{

a = p.getX();

b = p.getY();

}

void display() { cout << "a = " << a << " b = " << b << endl; }

};

int main()

{

Item i1;

Product p1;

p1.setData(3, 4);

i1 = p1;

p1.display();

i1.display();

return 0;

}

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

#include <iostream>

using namespace std;

class Item

{

int a, b;

public:

Item() {}

Item(int x, int y)

{

a = x;

b = y;

}

void display() { cout << "a = " << a << " b = " << b << endl; }

};

class Product

{

int x, y;

public:

void setData(int a, int b)

{

x = a;

y = b;

}

int getX() { return x; }

int getY() { return y; }

operator Item()

{

Item i(4, 3);

return i;

}

void display() { cout << "x = " << x << " y = " << y << endl; }

};

int main()

{

Item i1;

Product p1;

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.

#include <iostream>

using namespace std;

class Invent1

{

private:

int x, y;

public:

Invent1() {}

Invent1(int x, int y)

{

this->x = x;

this->y = y;

}

operator float()

{

return x + y;

}

int getX() { return x; }

int getY() { return y; }

void display() { cout << "x = " << x << " y = " << y << endl; }

};

class Invent2

{

private:

int a, b;

public:

Invent2() {}

Invent2(int a, int b)

{

this->a = a;

this->b = b;

}

Invent2(Invent1 i1)

{

a = i1.getX();

b = i1.getY();

}

void display()

{

cout << "a = " << a << " b = " << b << endl;

}

};

int main()

{

Invent1 i1(3, 4);

Invent1 i2;

float f;

f = i1; // invent1 to float

i2 = i1; // invent1 to invent2

cout << "Float f = " << f << endl;

i2.display();

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.

#include <iostream>

using namespace std;

class Time

{

private:

int hrs, minute, second;

public:

Time() {}

Time(int d)

{

hrs = d / 3600;

d = d % 3600;

minute = d / 60;

second = d % 60;

}

void display() { cout << "Hours = " << hrs << ", Minute = " << minute << ", second = " << second << endl; }

};

int main()

{

int duration;

cout << "Enter duration in seconds: ";

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.

#include <iostream>

using namespace std;

class Time

{

private:

int hrs, minute;

public:

Time() {}

Time(int hrs, int minute)

{

this->hrs = hrs;

this->minute = minute;

}

int getHours() { return hrs; }

int getMinute() { return minute; }

void display() { cout << "Hours - " << hrs << " Minute - " << minute << endl; }

};

class Minute

{

private:

int minute;

public:

Minute() { minute = 0; }

Minute(Time t)

{

cout << "Constructro() called " << endl;

minute = t.getMinute();

}

void display() { cout << " Minute - " << minute << 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.

#include <iostream>

using namespace std;

class Rupee

{

private:

int rupee;

public:

Rupee() { rupee = 0; }

Rupee(int r) { rupee = r; }

operator int()

{

return rupee;

}

};

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.

#include <iostream>

using namespace std;

class Dollar

{

private:

int dollar;

public:

Dollar() { dollar = 0; }

Dollar(int r) { dollar = r; }

void display() { cout << "Dollar = " << dollar << endl; }

};

int main()

{

int x = 50;

Dollar d;

d = x;

d.display();

return 0;

}

#include <iostream>

using namespace std;

class Dollar

{

private:

int dollar;

public:

Dollar() { dollar = 0; }

Dollar(int r) { dollar = r; }

void display() { cout << "Dollar = " << dollar << 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.

#include <iostream>

using namespace std;

class Dollar;

class Rupee

{

private:

float rupee;

public:

Rupee() { rupee = 0; }

Rupee(float r) { rupee = r; }

float getRupee() { return rupee; }

void display() { cout << "Rupee = " << rupee << endl; }

};

class Dollar

{

private:

float dollar;

public:

Dollar() { dollar = 0; }

Dollar(Rupee r) { dollar = r.getRupee() / 82.0; }

float getDollar() { return dollar; }

operator Rupee()

{

return dollar \* 82;

}

void display() { cout << "Dollar = " << dollar << 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;

}