Q1. 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 {

	private:
		int real, imag;

	public:

		void operator =(int x) {

			real = x;
			imag = x;
		}

		void disp(void) {

			cout << real << " + " << imag << "i";
		}


};


int main() {

	Complex c1;

	int x = 5;

	c1 = x;
	c1.disp();
}
```

Q2. 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 {
	private :
		int real, imag;

	public:
		void setData (int real, int imag);

		operator int() {

			return (real + imag);
		}
};

void Complex::setData(int real, int imag) {

	this->real = real;
	this->imag = imag;

}

int main() {

	Complex c1;

	c1.setData(3, 4);

	int x;

	x = c1;

	cout << x;

}
```

Q3. 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 {

	private:
		int a, b;

	public:

		void setData(int a, int b) {

			this->a = a;
			this->b = b;
		}
		int getA() {
			return a;
		}
		int getB() {
			return b;
		}

};

class Item {

	private:
		int z;

	public:
		void disp() {

			cout << "z : " << z ;
		}
		Item() {
		}
		Item(Product p1) {
			z = p1.getA() + p1.getB();
		}
};
int main() {

        Item i1;
        Product p1;
        p1.setData(3, 4);
        i1 = (Product)p1;
        i1.disp();

}
```

Q4. Create a 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 {

	private:
		int z;

	public:
		void disp() {

			cout << "z : " << z ;
		}

		Item() {
		}

		Item(int x) {
			z = x;
		}
};

class Product {

	private:
		int a, b;

	public:

		void setData(int a, int b) {

			this->a = a;
			this->b = b;
		}

		operator Item() {

			Item i = (a + b);
			return i;
		}

};

int main() {

	Item i1;
```

```
        Product p1;
        p1.setData(3, 4);
        i1 = p1;
        i1.disp();

}
```

Q5. 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 {

	private:

		int a, b;

	public:

		Invent1() {
		}
		Invent1(int a, int b) {
			this->a = a;
			this->b = b;
		}

		operator float() {
			return (a + b) * 1.0;
		}

		int getA() {
			return a;
		}
		int getB() {
			return b;
		}

		void disp() {
			cout << "Invent1 a : " << a << " b : " << b << endl;
		}
};

class Invent2 {

	private:

		int a, b;

	public:
		Invent2() {}
```

```
                    Invent2(Invent1 z) {
                              a = z.getA();
                              b = z.getB();
                    }

                    void disp() {
                              cout << "Invent2 a : " << a << " b : " << b << endl;
                    }
};

int main() {
          Invent1 s1(4, 5);
          Invent2 d1;
          float tv;
          tv = s1;
          d1 = s1;

          cout << "float tv : " << tv << endl;
          s1.disp();
          d1.disp();



}
```

Q6. 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>
#include <string.h>

using namespace std;

class Time {

	private:
		int second;

	public:

		Time(int duration) {

			second = duration * 60;
		}

		void disp() {
			cout << "seccond : " << second;
		}
};

int main() {

	int duration;
	cout << "Enter time duration in minutes" << endl;
	cin >> duration;
	Time t1 = duration;
	t1.disp();


}
```

Q7. 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;
}
```

```
#include <iostream>
using namespace std;

class Time {

	private:
		int H, M;

	public:

		Time(int h, int m) {

			H = h;
			M = m;
		}

		void display(void) {

			cout << H << " Hr : " << M << " min" << endl;
		}

		int getH(void) {

			return H;
		}

		int getM(void) {

			return M;
		}

};

class Minute {
	private:
		int min;

	public:
		Minute() {

			min = 0;
		}

		void display(void) {

			cout << min << " min" << endl;
		}

		Minute(Time t) {

			min = t.getH() * 60;
			min = min + t.getM();

		}


};

int main() {

	Time t1(2, 30);
	t1.display();

	Minute m1;
	m1.display();

	m1 = t1;

	m1.display();
	t1.display();
}
```

Q8. 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 {

	private:
		int rs;

	public:

		Rupee(int rs) {

			this->rs = rs;
		}

		operator int() {

			return rs;
		}
};

int main() {

	Rupee r = 10;
	int x = r;
	cout << x;
	return 0;
}
```

Q9.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 {

	private:
		int d;

	public:

		Dollar() {
		}

		Dollar(int doller) {
			d = doller;
		}

		void display(void) {
			cout << d;
		}
};

int main() {

	int x = 50;
	Dollar d;
	d = x;
	d.display();
}
```

Q10.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 {

	private:
		float r;

	public:

		Rupee(float r) {

			this->r = r;
		}

		int getR(void) {

			return r;
		}

		void display(void) {

			cout << "Rupee : " << r << endl;
		}
};


class Dollar {

	private:
		float d;

	public:
		Dollar() {
		}

		Dollar(Rupee rs) {

			d = rs.getR() / 80.0;
		}
```

```
        void display(void) {

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

        operator Rupee() {

            return d * 80.0;
        }
};


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();
}
```