Lab – 7

1. Create a class with one private integer variable and public constructors along with show() function to display the variable value. Implement the operator overloading program using the following member functions:

   (a) No return just increment the object in the main function e.g. obj++;

```
void operator ++() {
//…
}
```

   (b) Return object after increment in the main function e.g. obj2 = obj1++;

```
MyClass operator ++() {
//…
}
```

2. Implement the binary operator overloading to implement addition of a distance object consisting two integer variables x,y. d3 = d1+d2; operations in main and member function syntax shown below.

```
Distance operator+(Distance& d) {
Distance d3;
d3.x = x+d.x; d3.y = y+d.y;
return d3;
}
```

3. Write a program to overload unary minus(-) using the given member function

```
Distance operator- () {
    feet = -feet;
    inches = -inches;
    return Distance(feet, inches);  // or return *this;
  }
```

4. Practice to overload the following operators:

- Arithmatic (+ , – , * , /)
- relational (== or <= etc)
- logical (&& or || etc.)

Hint:

```
bool operator < (const Distance& d) {
      if(feet < d.feet) {
        return true;
      }
```

5. Overload '+' operator using the following code:

```
Complex operator + (Complex const &obj) {
        Complex c;
        c.real = real + obj.real;
        c.imag = imag + obj.imag;
        return c;
    }
```

6. Create a class Time with three private variables int h,m,s; Create a function to overload '+' operator to add two time variables.

```
int main(){
   Time t1(5,15,34),t2(9,53,58),t3;
   t3 = t1 + t2;  t3.show();
}
```

7. Write a program for operator overloading using friend function using the following code:

```
class Test{
//…
public:
 friend void operator - (Test &x);
};
void operator-(Test &x){
//…
}
int main(){
Test x1;
-x1;
}
```

8. Write a program to convert basic data type (float) to user defined data type (object).

```
class Test {
private: //….
public:
Test ( data_type) { // conversion code }
};
```

9. Write a program to convert UDT to basic data type (float)

```
class Test{
```

```
 public:
operator data_type() { //Conversion code }
};
```

10. How will you convert one UDT to another UDT. For example conversion of polar to cartesian system.

```
Polar p(10,5);
Cartesian c = p;
c.show();
```