Operator Overloading

“Syntactic Sugar”

Prerequisite: None

Md. Saidul Hoque Anik
onix.hoque.mist@gmail.com

The idea of Operator Overloading

```
int main()
{
    int x = 10;
    int y = 20;









}
```

The idea of Operator Overloading

```
int main()
{
    int x = 10;
    int y = 20;

    int z = x + y;




}
```

The idea of Operator Overloading

```
int main()
{
    int x = 10;
    int y = 20;

    int z = x + y;

    Point p1(1, 1);
    Point p2(2, 2);




}
```

The idea of Operator Overloading

```
int main()
{
    int x = 10;
    int y = 20;

    int z = x + y;

    Point p1(1, 1);
    Point p2(2, 2);

    Point p3 = p1 + p2;

    p3.display();
}
```

The idea of Operator Overloading

```
int main()
{
    int x = 10;
    int y = 20;

    int z = x + y;

    Point p1(1, 1);
    Point p2(2, 2);

    Point p3 = p1 + p2;

    p3.display();
}
```

Point Class

```
class Point
{
    int x, y;
public:
    Point(int x, int y)
    {
        this->x = x;
        this->y = y;
    }
    void display()
    {
        printf("(%d, %d)\n", x, y);
    }
```

Point Class (cont.)

```
    Point operator+(Point p2)
    {
        Point ret(x + p2.x, y + p2.y);
        return ret;
    }
};
```

Exercise

1. Design a ComplexNumber class which will have two members: real and imaginary

2. Write necessary code so that the following snippet works.

```
int main()
{
   ComplexNumber c1(10, 10), c2(20, 20);

   ComplexNumber sum = c1 + c2;
   ComplexNumber prod = c1 * c2;
}
```

$$(a_1 + b_1 i) \cdot (a_2 + b_2 i) = (a_1 a_2 - b_1 b_2) + (a_1 b_2 + a_2 b_1) i$$

Writing operator function outside class

```
        printf("(%d, %d)\n", x, y);
    }

    friend Point operator+(Point p1, Point p2);
};

Point operator+(Point p1, Point p2)
{
    Point ret(p1.x + p2.x, p1.y + p2.y);
    return ret;
}
```

Task

Write the add an multiplication of complex number as non-member function.

A few things to note

1. For operator overloading to work, at leas one of the operands must be a user defined class object.

2. Any constructor that can be called with a single argument works as a conversion constructor, means it can also be used for implicit conversion to the class being constructed.

Reference

https://www.geeksforgeeks.org/operator-overloading-c/amp/