

C, C++, DSA in depth

## Operator overloading



Saurabh Shukla (MySirG)

## Agenda

- ① Recall add method
- ② Operator overloading
- ③ Polymorphism
- ④ Not all the operators can be overloaded
- ⑤ Overloading of binary operator
- ⑥ Overloading of unary operator

## Recall add method

```
Complex Complex :: add (Complex C)
{
    Complex temp;
    temp.a = a + C.a;
    temp.b = b + C.b;
    return temp;
}
```

$c_3 = c_1.add(c_2);$

change name from add to +

---

```
Complex Complex :: Operators + (Complex C)
{
    Complex temp;
    temp.a = a + C.a;
    temp.b = b + C.b;
    return temp;
}
```

$C_3 = C_1 \cdot \text{operator} + (C_2);$   
OR

$C_3 = C_1 + C_2;$

# class

---

- ① variables
- ② functions
- ③ operators

cout << x

cout . operator<<(x)

object . variables

object . functions( )

object . operator + (arg)

object + arg

# Operator Overloading

When one operator symbol is overloaded with multiple operations, it is known as overloaded operator.

Defining an operator in a class, is providing a new behaviour of operator for specific type operands.

In simple words, operator overloading occurs when you define an operator with respect to the class.

## Polymorphism

Operator Overloading is another way of implementing Polymorphism.

$3 + 4$       int + int

$5.4 + 6.3$       double + double

$C1 + C2$       complex + complex

$t1 + t2$       Time + Time

# Not all the operators can be overloaded

- Only those symbols can be defined as an operator which were valid operators in C language.
- There are few operators in C language which you cannot overload in C++
  - `sizeof()` member access operator
  - `.` pointer to member operator
  - `*` conditional operator
  - `?:` scope resolution operator
  - `::`

## Overloading of binary operators

When a binary operator is overloaded in a class as a member, only left operand is a caller object and right operand is an argument.

$$C3 = C1 + C2$$

↑  
caller object

## Overloading of Unary Operators

The way of writing /using unary operator will remain same for overloaded version.

>

==

- .

\*

[ ]

( )

++ <--> post  
-- <--> pre

==

<<

>>