Operator Overloading, Friends, and References
Problem #1 Complex Number Class

• Create a class which express a complex number
• The form of a complex number is “[R] + i[I]”
• [R] is a real number, [I] is a imaginary number
• The class should be able to do arithmetic operation
  • Ex)
    • (1+i4) + (5+i2)
    • (3-i5) - (1+i4)
    • (3+i4) * (3-i3)
    • (2+i4) / (1+i3)
Problem #1 Complex Number Class

• Implement arithmetic operations by overloading operators

• operator=
• operator+, operator+=
• operator-, operator-=
• operator*, operator*= 
• operator/, operator/=
Problem #1 Complex Number Class

• Implement arithmetic operations by overloading operators

• operator=
• operator+, operator+=
• operator-, operator-=
• operator*, operator*=,
• operator/, operator/=,

Return new complex instance
Problem #1 Complex Number Class

• Implement arithmetic operations by overloading operators

• operator=
• operator+, operator+=
• operator-, operator-=
• operator*, operator*= 
• operator/, operator/= 

Modify the lvalue
Problem #1 Complex Number Class

• Complex class has 3 initializers
  • Complex(complex)
  • Complex(real, img) // Both real and img is double
  • Complex(str) // str is a form “[R] op i[I]”
    // op is + or −, [R] can be a minus value

• Complex class has functions as below
  • operator+, −, *, /, +=, −=, *=, /=, =
  • println():
    Print complex number, a form of “[R] op [I]i”
    After printing, print a new line character (“\n”)
Problem #2 Complex Number Class Ex

• For now, we can calculate operations such as
  • complex + complex
  • complex + str

• But, it isn’t allowed to calculate a form of
  • str + complex

• Solving this problem by using global functions
  • But real and img value of complex are private variables
  • How can we use this values from global functions?
Problem #2 Complex Number Class Ex

• Let’s add a feature about printing a value
• We want to use cout with << operation
  • ostream& operator<<(ostream&os, ~)
  • ex)

Code:
Complex a(1, 3);
cout << a << endl;

Output:
1 + 3i
Thank You