

Number

Sungchul Lee

Learning Objection

➤ Number

- ❑ Integer and Floating

➤ Arithmetic Computation

- ❑ Arithmetic Expression

- ❖ Addition: + $x+y$

- ❖ Subtraction: - $x-y$

- ❖ Multiplication: * $x*y$

- ❖ Division: / x/y

- ❖ Modulo: % $x\%y$

- ❖ Increase (++) / decrease operation (--)

Integer Number Type

➤ Ranges of Values for Integer

1. byte: -128 to 127
2. short: -32,768 to 32,767
3. int: -2,147,483,648 to 2,147,483,647
4. long: -9,223,372,036,854,775,808 to
9,223,372,036,854,775,807

➤ Example:

❑ `int age = 10;`

❑ `long countOfStar = 8764827384923849L;`

Floating Point Types

- float: 32 bits wide
 - approximately 7 decimal places of accuracy
 - range of values is approximately
 - ❖ -3.4×10^{38} to $+3.4 \times 10^{38}$.
- double: 64 bits wide
 - approximately 16 decimal places of accuracy
 - range of values is approximately
 - ❖ -1.8×10^{308} to $+1.8 \times 10^{308}$
- Example:
 - **float** pi = 3.14F;
 - **double** morePi = 3.14159265358979323846;

Arithmetic Expression

➤ Any expression involving **numerical values** is called arithmetic expression

❑ Integer and Floating data type can use arithmetic expression

➤ Arithmetic operators (java use the same operator)

❑ Addition: + $x+y$

❑ Subtraction: - $x-y$

❑ Multiplication: * $x*y$

❑ Division: / x/y

❖ Integer division: Get an integer quotient when both x and y are integers

❖ $23/20 = 1$, not 1.15

❑ Modulo(remainder): % $x\%y$

❖ Returns the remainder of a division ($7 \% 3 = 1$)

```
int a = 23;  
int b = 20;  
System.out.print(a/b);
```

Precedence in Expression

➤ How does you do the calculation?

❑ Let $x = 6$, $y = 3$;

❑ $z = x + 3 / y$

❑ What is the answer (z value)?

➤ Can you change it to Java code?

❑ `int x = 6, y = 3;`

❑ `int z = x + 3 / y;`

❑ `System.out.println(z);`

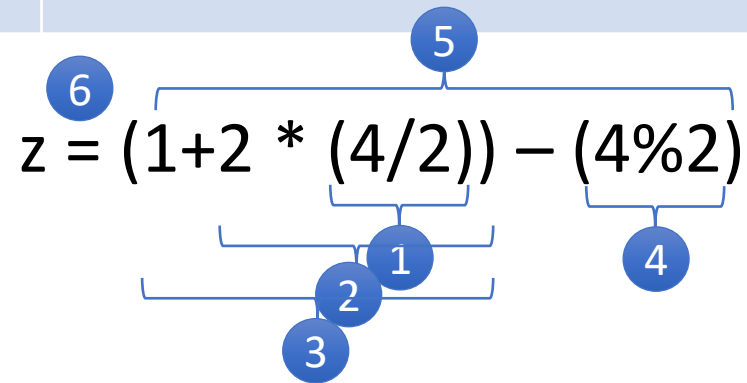
1. $3/y \rightarrow 1$
2. $x+1 \rightarrow 7$
3. $z = 7$



Precedence Rules

Operator	Operation	Order of evaluation (precedence)
()	Subexpression	Subexpressions are evaluated first. If parentheses are nested, the innermost subexpression is evaluated first. If two or more pairs of parentheses are on the same level, then they are evaluated from left to right
* / %	Multiplication Division remainder	Evaluated second. If there are several operators of this type, they're evaluated from left to right.
+ -	Addition subtraction	Evaluated next. If there are several operators of this type, they're evaluated from left to right.
=	Assignment	Evaluated last.

High Precedence



Assignment Statements with Arithmetic

- Variables can be reassigned as many times as you want!

`int x= 0;` //assign 0 into x ← Assignment Statements

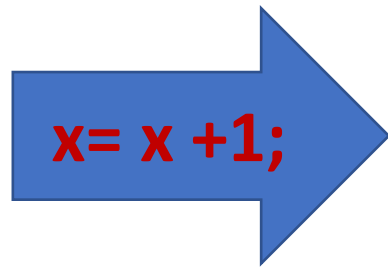
`x = 7;` // Assignment Statement

`x = x + 1;` // Assignment Statement

- When a variable changes, the old value is erased and a new one is written in.

❑ Before

❑ x 10



After

x 11

Increase (++) / decrease operation (--)

➤ Increase one using ++

□ Example: $x = x + 1$



➤ Decrease one using --

□ Example: $x = x - 1$



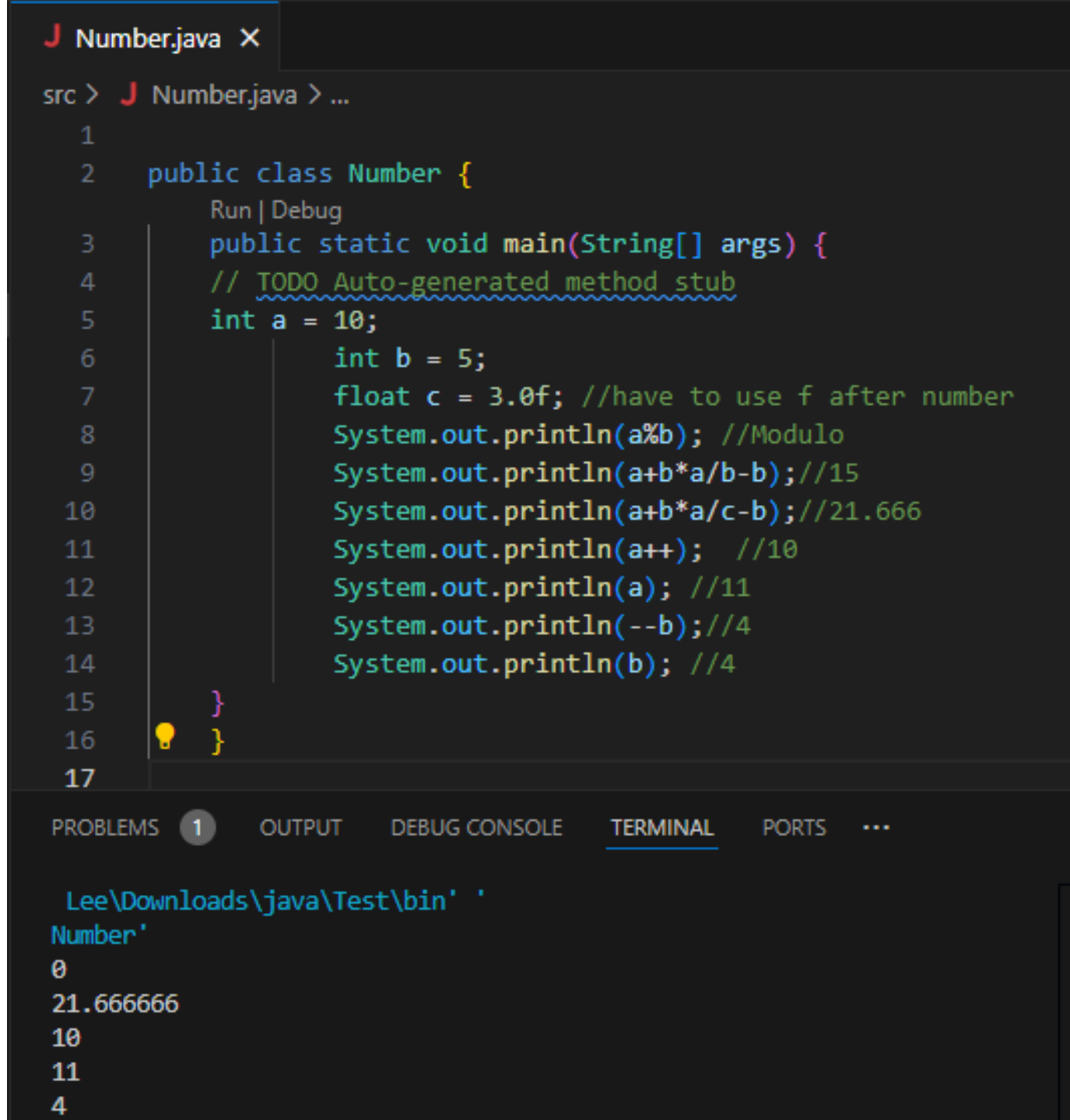
Practice

1. Make a new project (Reference: Create Project and Class File)
 - ☐ Project name: Number
2. Create a new Class File
 - ☐ Class name: Number
3. Coding:

```
public class Number {  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
        int a = 10;  
        int b = 5;  
        float c = 3.0f; //have to use f after number  
        System.out.println(a%b); //Modulo  
        System.out.println(a+b*a/b-b); //15  
        System.out.println(a+b*a/c-b); //21.666  
        System.out.println(a++); //10  
        System.out.println(a); //11  
        System.out.println(--b); //4  
        System.out.println(b); //4  
    }  
}
```

Practice – Code and Result

- Result



The screenshot shows an IDE with a Java file named `Number.java`. The code defines a `Number` class with a `main` method. The `main` method contains several `System.out.println` statements that output the results of various arithmetic and increment/decrement operations. The IDE's terminal window shows the output of the program, which matches the expected results from the code.

```
src > J Number.java > ...  
1  
2 public class Number {  
3     Run | Debug  
4     public static void main(String[] args) {  
5         // TODO Auto-generated method stub  
6         int a = 10;  
7         int b = 5;  
8         float c = 3.0f; //have to use f after number  
9         System.out.println(a%b); //Modulo  
10        System.out.println(a+b*a/b-b); //15  
11        System.out.println(a+b*a/c-b); //21.666  
12        System.out.println(a++); //10  
13        System.out.println(a); //11  
14        System.out.println(--b); //4  
15        System.out.println(b); //4  
16    }  
17 }
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS ...

```
Lee\Downloads\java\Test\bin' '  
Number'  
0  
21.666666  
10  
11  
4
```

Summary

➤ Number

❑ Integer and Floating

➤ Arithmetic Computation

❑ Arithmetic Expression

❖ Addition: + $x+y$

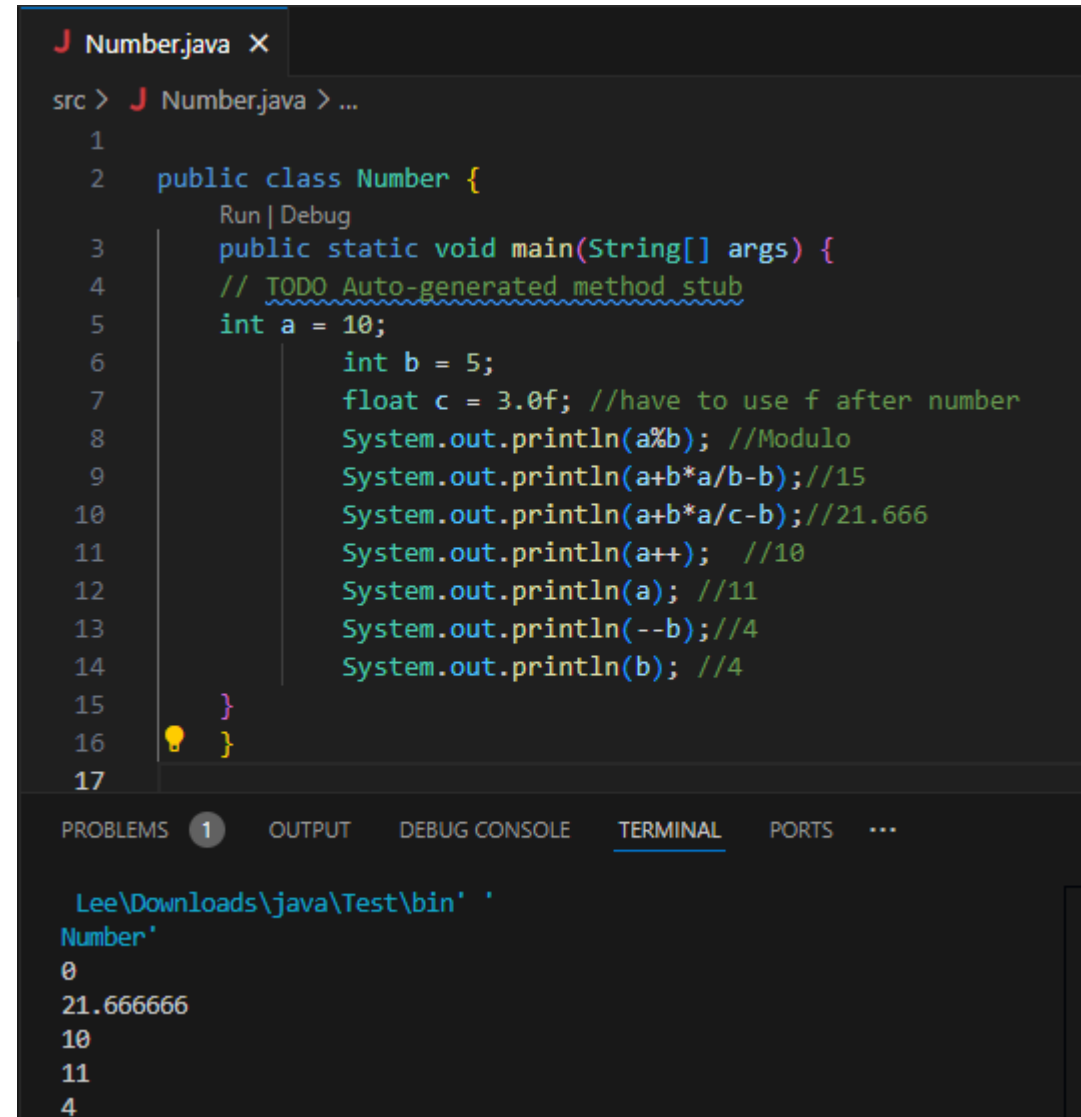
❖ Subtraction: - $x-y$

❖ Multiplication: * $x*y$

❖ Division: / x/y

❖ Modulo: % $x\%y$

❖ Increase (++) / decrease operation (--)



The screenshot shows an IDE with a file named `Number.java`. The code defines a `Number` class with a `main` method. The `main` method contains several lines of code that calculate and print values. The terminal output shows the results of these calculations.

```
src > J Number.java > ...
1
2 public class Number {
3     Run | Debug
4     public static void main(String[] args) {
5         // TODO Auto-generated method stub
6         int a = 10;
7         int b = 5;
8         float c = 3.0f; //have to use f after number
9         System.out.println(a%b); //Modulo
10        System.out.println(a+b*a/b-b); //15
11        System.out.println(a+b*a/c-b); //21.666
12        System.out.println(a++); //10
13        System.out.println(a); //11
14        System.out.println(--b); //4
15        System.out.println(b); //4
16    }
17 }
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS ...

```
Lee\Downloads\java\Test\bin' '
Number'
0
21.666666
10
11
4
```