2024.03.11

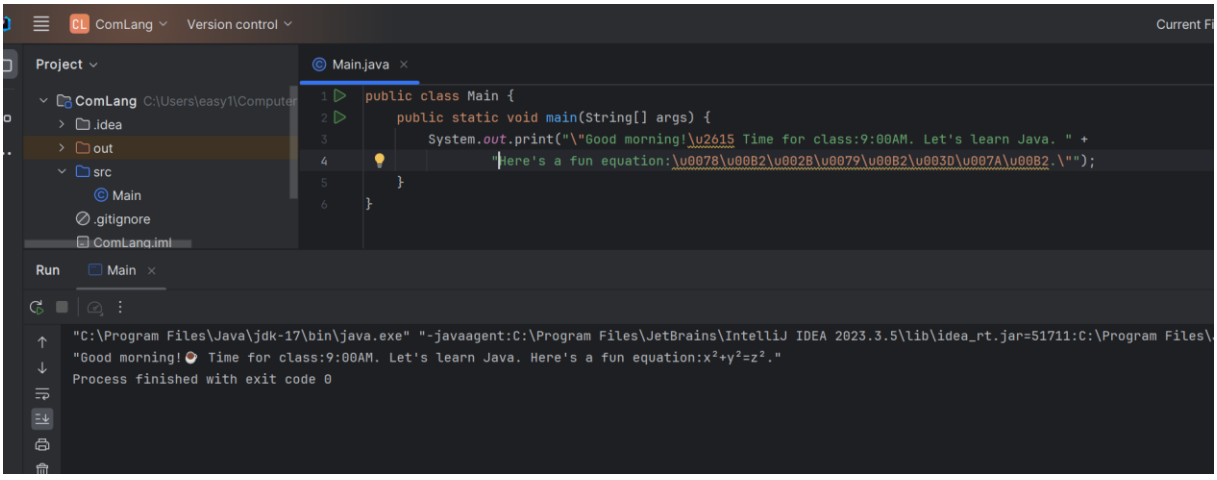
HOMEWORK 1

1. For each variable declaration case listed below, provide one valid variable name and one invalid variable name. Additionally, explain why the invalid variable name cannot be used (do not repeat invalid name reasons).

a. The class of a car driver.
   Valid: driverClass
   Invalid: driver Class
   Reason: Whitespaces are not allowed
b. The sea level of a region.
   Valid: seaLevel
   Invalid: abstract
   Reason: Java keywords are not allowed as variable name.
c. The temperature reading of a greenhouse sensor.
   Valid: temperature
   Invalid: temper@ature
   Reason: Special characters like @ are not allowed as variable name
d. The salary of an employee.
   Vaild: salary
   Invaild: true
   Reason: boolean literal(true or false) is not allowed

2. Write a program to print the following string using UNICODE literals (only for symbols). Capture your source code and the output of your program and paste them in this document. Double quotes should note appear in the output.

"Good morning! ☕ Time for class: 9:00 AM. Let's learn Java. Here's a fun equation: $x^2 + y^2 = z^2$."



3. Execute the following program. What is the output of this code? Explain why such a result was calculated.

```
public class Hello {
    public static void main(String[] args) {

        int intValue = 12345;
        System.out.println((byte) intValue);

    }
}
```

Output: 57
The statement System.out.println refers to the printing of an integer variable named intValue in byte form. Bites can be expressed in a range between -128 and 127, but 12345 is out of this range. If 12345 is binary, 0011 0000000000000011 1001. Bites are 8 bits, so the first 8 bits are thrown away in a binary representation of 12345 that requires 16 bits. Only 00111001 remains, and if we convert this value to decimal, it is 57.

4. How to declare a variable for the following case?

- Declare an int variable named age.
- Declare a float variable weight initialized to 70.5f.
- Declare a float variable totalWeight initialized with the sum of age and weight.
- Declare a char variable initial initialized with 'J'.
- Declare a final float constant ideal_temp initialized to 37.0f.

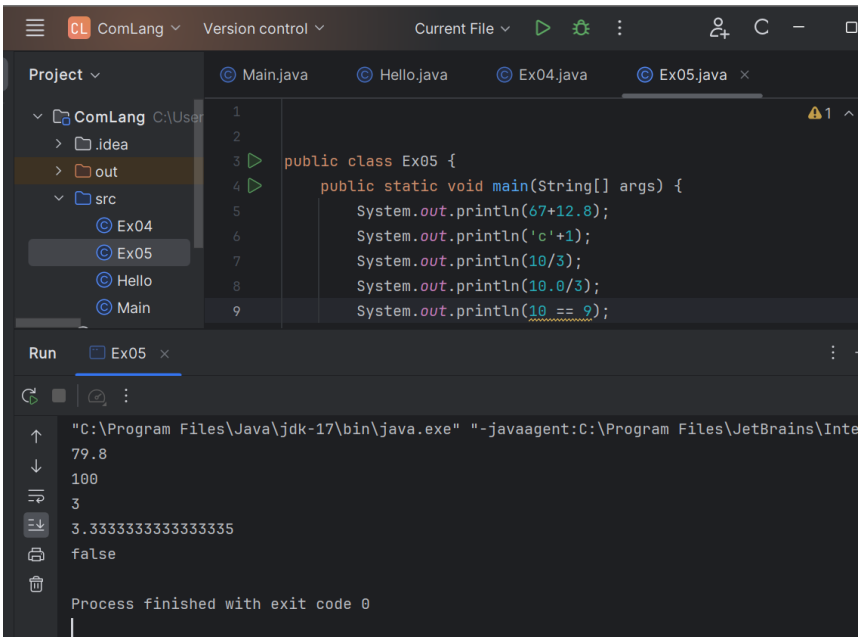
```
public class Ex04 {
    public static void main(String[] args) {
        int age;
        float weight = 70.5f;
        float totalWeight = age + weight;
        char initial = 'J';
        final float ideal_temp = 37.0f;

    }
}
```

A compilation error will occur because the age variable used when initializing totalWeight was not initialized. Initializing the age variable will resolve this error.

5. What is the result and type of the following statements? Explain the reason as well.

- 67+12.8
- ‘c’+1
- 10/3
- 10.0/3

- 10 == 9



- 67+12.8: 67 is integer and 12.8 is floating-point numbers. Java performs operations by converting it into floating-point numbers with a larger range. 67 is automatically converted to 67.0, calculated, and the corresponding value is double.
- 'c' + 1: Characters are represented by ASCII values, where c corresponds to 99. Therefore, the value is 100 and int type.
- 10/3: 10/3 is the division of integers. The result is calculated as an integer and the rest is discarded. The result is 3 and int.
- 10.0/3: 10.0/3 is the division of floating-point numbers and integers. Like the reason for the first result, the result is a floating-point number. It has double form.
- 10 == 9: 10 == 9 compares whether two integers are the same. Since the values are not the same, the result is false and boolean.

6. Execute the following codes. Then, you will see eight values printed on the console. For each value, explain how it was calculated.

```
public class Ex07 {
    public static void main(String[] args) {
        byte b = 127;
        int i = 100;

        System.out.println(b + i);
        System.out.println(10 / 4);
        System.out.println(10.0 / 4);
        System.out.println((char) 0x12340041);
        System.out.println((byte) (b + i));
        System.out.println((int) 2.9 + 1.8);
```

```
        System.out.println((int) (2.9 + 1.8));
        System.out.println((int) 2.9 + (int) 1.8);

    }
}
```

1. b + i : b is 127 and i is 100, so these two values add up to 227.

2. '10/4': Since it is the division of two integers, the remaining part is discarded, resulting in 2.

3. 10.0/4: The division of floating-point numbers and integers. The result is in the form of floating-point. The result is 2.5.

4. (char) 0x12340041: 0x12340041 is transformed to output the corresponding Unicode character. The corresponding Unicode character is A.

5. (byte)(b+i): Cast the result of b+i in bytes; overflow occurs because the result of b+i is 227 outside the range of bytes. Result is -29.

6. (int) 2.9 + 1.8: 2.9 into int gives 2. Add 1.8 to 2 and the result is 3.8.

7. (int) (2.9 + 1.8): Add 2.9 and 1.8 to get 4.7 and transform it into an integer to get 4.

8. If (int) 2.9 + (int) 1.8: 2.9 is transformed into an integer, 2 and 1.8 are transformed into 1, and if the two are added, it becomes 3.