AST10106

Introduction to Programming JAVA

Semester A 2015/16

Lab05

Variables, Arithmetic Operators, Expressions and Assignment

Submission Details

In this lab, you are required to write **TWO** Java programs to solve the given problems shown in the section “Lab Exercises”. To make the program implementation easier, you are suggested to write your program using **eclipse** instead of doing it directly using paper & pencil. After you have completed the implementation using **eclipse**, submit your program files (i.e. WeightProgram.java & QuadraticCalculator.java) to the assignment page of Canvas. The deadline would be **ONE week** after your lab session conducted. You are reminded to double-check your solution to verify everything in correct before submission.

Note: You are required to put your name, student ID and lab section at the beginning of your source file.

Note: Only answer of selected question will be graded.

Objective

The objective of this lab is to reinforce your programming concepts (e.g. basic Java input / output, variable declarations and use of operators) taught in the last few lectures and labs.

Software Required

The following software is required for this lab.

1. Java Development Kit, eg JavaSE 1.8
2. An Integrated Development Environment, e.g. eclipse

Introduction

In this lab, you will practice:

1. Forming a program skeleton by declaring a class with a main method / operation
2. Declaring variables to store values in Java programs
3. Obtaining inputs from keyboard using methods / operations in the Scanner class
4. Solving problems by performing arithmetic operations (+, -, *, /, %) on inputs
5. Displaying the solution of problems on screen using methods / operations in the System class

Background

1. Form a program skeleton by defining a class with a main method / operation

- **Syntax:**

```
// The filename of the Java source file should be exactly the same
// as the <class name>
public class <class name>
{
    // Define a main method / operation, which is the starting point of the program
    public static void main(String[] args)
    {
        // Program statements here
    }
}
```

2. Declare variables for storing values

- **Syntax:**

```
// Declare a variable with name <name of variable> in type <type>
<type> <name of variable>;
// Declare a variable with name <name of variable> in type <type> with
// initial value <initial value>
<type> <name of variable> = <initial value>;
// Declare a number of variables with name <variable name1>,
// <variable name2>, … all in type <type>
<type> <variable name1>, <variable name2>, …;
// Declare a number of variables with name <variable name1>,
// <variable name2>, … all in type <type> with initial value <initial value1>,
// <initial value2>, …, respectively
<type> <variable name1> = <initial value1>, <variable name2> = <initial value2>, …;
```

where <type> could be in one of the following types:
byte, short, int, long, float, double, char, boolean

3. Obtain inputs from keyboard and store them in variables:

- **Syntax:**

```
// Get the class definition of the Scanner class from java.util package
import java.util.Scanner;
// Create a Scanner object with name <object name>
// System.in is the "standard" input stream, which is keyboard
Scanner <object name> = new Scanner(System.in);
// Use the Scanner class's operations (methods) to read input
<variable name> = <object name>.<method name>();
```

A brief summary of Scanner class operations:

Method name	**Description**
boolean nextBoolean()	Returns the next input token as a boolean value
byte nextByte()	Returns the next input token as a byte value
short nextShort()	Returns the next input token as a short value
int nextInt()	Returns the next input token as a int value
long nextLong()	Returns the next input token as a long value
float nextFloat()	Returns the next input token as a float value
double nextDouble()	Returns the next input token as a double value
String next()	Returns the next input token as a String value
String nextLine()	Returns all input remaining on the current line as a String value

4. Perform calculation using arithmetic operations:

- **Syntax:**

```
// Add the two values stored in variable 1 and variable 2
<name of variable for the result> = <name of variable 1> + <name of variable 2>;
// Subtract the value stored in variable 2 from variable 1
<name of variable for the result> = <name of variable 1> - <name of variable 2>;
// Multiply the value stored in variable 1 by the value stored in variable 2
<name of variable for the result> = <name of variable 1> * <name of variable 2>;
// Divide the value stored in variable 1 by the value stored in variable 2
<name of variable for the result> = <name of variable 1> / <name of variable 2>;
// Find the remainder of value stored in variable 1 divided by the value stored in
// variable 2
<name of variable for the result> = <name of variable 1> % <name of variable 2>;
```

5. Display information on screen:

- **Syntax:**

```
// Print data on screen and the cursor will stay at the end of the string printed
System.out.print(<data> [+ <data>]);
// Print the data on screen and move the cursor to the beginning of the next line
System.out.println(<data> [+ <data>]);
// Print nothing
System.out.print(“”);
// Print nothing, but move the cursor to the beginning of the next line
System.out.println(“”);
```

Lab Exercises

In this lab, you will be asked to solve **TWO** problems using Java. For these two problems, write Java programs from the following specifications of the problems.

Question 1:

Write a Java program (class name: WeightProgram) to find the mass of fruits bought by customers. The mass can be found by balancing the fruits bought with the set of weights available below.

Set of weights: 100g, 50g, 20g, 10g, 5g and 1g.

Notice that the program should first read the amount of fruits to buy, such as 5 apples (mass = 103 each) and 2 mangos (110 each) and then find the **MINIMUM NUMBER OF WEIGHTS** used. Your program should produce input/output that looks like the following:

(Note: The underlined text refers to the user input. Also, please follow closely to the output format shown below.)

Sample Output:

```
Weight program
Enter number of apples to buy: 5
Enter number of mangos to buy: 2
100g-weight: 7
50g-weight: 0
20g-weight: 1
10g-weight: 1
5g-weight: 1
1g-weight: 0
```

Question 2:

A quadratic equation is an equation defined as a polynomial of degree two, it could be represented by: $ax^2 + bx + c = 0$,
where $a \neq 0$, b, c are called coefficient of the equation and x is the unknown value we would like to find. To solve the quadratic equation, we can apply the following formula:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Write a Java program (class name: QuadraticCalculator) to compute the solution(s) of a user specified quadratic equation.

The program should first welcome the user by printing a welcome banner:

```
Welcome to use Quadratic Calculator
```

Starting on the next line, the program should prompt the user to input the coefficients of the quadratic equation: a, b and c. It will prompt the user by printing:

```
Please enter the value of a: 1
Please enter the value of b: 3.5
Please enter the value of c: 2
```

The user will input the corresponding value after the print line.

Assumptions:

1. Notice that the number inputted by the user can be assumed to be correct (this lab does not require input validation). Therefore, you can simply assume that user won't input some invalid numbers, like "a", "abc", "-12-0.-2" etc.
2. The value b^2-4ac is always greater than or equal to 0, i.e. negative value doesn't appear.

Your program should print out solutions according to the given method:

(Note: The underlined text refers to the user input. Also, please follow closely to the output format shown below.)

Sample Output:

A session of inputs and outputs:

```
Welcome to use Quadratic Calculator
Please enter the value of a: 1
Please enter the value of b: 3.5
Please enter the value of c: 2
x1 = -0.7192235935955849
x2 = -2.7807764064044154
```

Another session of inputs and outputs:

```
Welcome to use Quadratic Calculator
Please enter the value of a: 1
Please enter the value of b: 2
Please enter the value of c: 1
x1 = -1.0
x2 = -1.0
```

Marking Scheme

The marking of this exercise will be based on the following criteria.

Graded items	Weighting
1. Correctness of program (i.e. whether your code is implemented in a way according to the requirements as specified.)	60%
2. Presentation of the Java codes (i.e. whether the program is properly indented, how close you follow the common conventions as mentioned in class, etc.)	30%
3. Documentation (with reasonable amount of comments embedded in the code to enhance the readability)	10%
	100%

Program Submission Checklist

Before submitting your work, please check the following items to see you have done a decent job.

Items to be checked ☑ / ☒

1. Did I put my name, student ID and lab section at the beginning of all the source files? □
2. Did I put reasonable amount of comments to describe my program? □
3. Are they all in .java extension and named according to the specification? □
4. Have I checked that all the submitted code are compliable and run without any errors? □
5. Did I zip my source files using Winzip / zip provided by Microsoft Windows? Also, did I check the zip file and see if it could be opened? ***(Only applicable if the work has to be submitted in zip format.)*** □
6. Did I submit my lab assignment to blackboard? □