CS 250 - Fall 2021 - Lab 01: Review CS234

Available: Aug 25, 2021.
Due date: **Aug 30, 2021, 11:59 PM**

Objectives:

The objectives of this lab are:

- Familiarizing with the Eclipse Programming Environment.
- Practice editing, compiling, running and revising a Java application.

Prerequisite:

- Read Chapter 0 to 7 of Java, Java, Java, 3E and read through this document.
- Install JDK 8 and Eclipse following the links in the Software section at Resources JDK 8 Installation instructions

Part 1: Install Java SDK 8 and Eclipse

Eclipse is an integrated development environment (IDE) for Java, C++, and other languages. It combines software resources for editing, compiling and running programs. In this IDE Java programs consist of two parts:

- the Java code itself, which contains the text of your source program and
- a project, which contains the Java bytecode that will be executed when you run your program.

The Eclipse Java project combines precompiled code from the Java class libraries with the code compiled from your Java source program, thereby increasing the functionality of your code. This lab will take you through the steps required to edit, compile, and run Java programs.

Close all applications that may be running on your computer in order to make as much memory as possible available to Eclipse.

Part 2: Variables and Printing to Console

Step 1. Run the Eclipse IDE

NOTE: Remember how to find this program because you're going to need it every week.

Step 2. Create the Lab01 project.

Once you have Eclipse running, create a new project and configure it according to the following directions:

- Click on the "Create a java project" link in the Package Explorer window or right click on that window then select NEW then Java project. Enter the name of your project, i.e. "Lab01". The "Location" text box should indicate that your project will be stored on the desktop. In the project JRE setting, make sure at least one Java SE version is selected, default to JavaSE-1.8 if you followed instructions on the Prerequisites section. Click on Finish to create the project.
- In the Package Explorer window, right click on the "src" item, then select "New" and then Class. Next, create the P1 java class with a static main method. This class will handle the requirements of the main application in this lab.

Step 3. Implement and test the first Java application.

1. Declare the following variables in the first program:

- Two integer variables i0 and i1 with values 12345 and -99999 correspondingly
- Two float variables f0 and f1 with values 1.2345F and 99.9999F correspondingly
- Two double variables d0 and d1 with values 0.0001 and 12345.98765

2. Using System.out.println() method to print the following expressions:

- (i0 + i1) / 10
- (i0 + i1) / 10.0
- i0 % 100
- (i0 - 2345) * 10
- i0 - 2345 * 10
- (f0 + f1) / (d0 * d1)
- 0.1 + 0.2 - 0.3
- 6 % 4 + 12 - 3 * (8 + 3) / 2

3. Try with other print methods with formatting:

- Using System.out.println to print (i0 + “,” + i1)
- Using System.out.printf to print (“i0 + i1 = %d\n”, i0 + i1)

- Using System.out.printf to print ("d0 + d1 = %.2f\n", d0 + d1)
- Using System.out.printf to print ("d0 + d1 = %.5f\n", d0 + d1)

Part 3: Review the Scanner class

Scanner class is a useful class of Java to read user inputs. Some of the popular methods in this class are described below:

- nextInt() - reads an integer
- nextDouble() - reads a double
- next() - reads a token
- nextLine() - reads a line of text
- hasNext() - checks if there is another token
- close() - closes the Scanner object

In the third part of this lab, we will solve the quadratic equation $ax^2 + bx + c = 0$ to find two roots: positive and negative roots. You need to use Scanner to read a, b, c inputs from the users, solve the equations, and display the required two roots.

The following steps guide you through implementing the code for this part:

1. Declare three variables a, b, c to represent the quadratic coefficients either using float or double

2. Create an instance of Scanner to read from the keyboard - the variable name is up to you

3. Using System.out.print to prompt the user to enter the three coefficients sequentially and use the Scanner object to read the user inputs into a,b,c:

    "Enter the coefficient a: "

    "Enter the coefficient b: "

    "Enter the coefficient c: "

4. Display the quadratic formula with the three inputs a, b, c of the user with the following format:

    “The input formula: $ax^2 + bx + c = 0$”

Note: You need to replace a, b, c with the values inside three variables. For example if a=1, b = -11, c = 24, you need to display $1x^2 - 11x + 24 = 0$

5. Solve the above equation to find the positive and negative root and display them to the Console. The equation to find the roots is displayed below:

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

Note: you are required to print two roots with two digits after the decimal points, i.e., 1.23

Run your program with different inputs to test the validity.

Part 4: Review methods and conditional statement

In the fourth part of this lab, revise the use of methods and conditional statements.

The following steps guide you through implementing the code for this part:

1. Define and implement a method named compareInt() in your Java class to compare and print the comparison between two input integer parameters. Use the following signature for this method:

```
/**
 * This method compares two input integers a and b
 *    and display the comparison result between them
 * @param a, b: the input numbers
 * @return: None but print to console the comparison result
 */
public static void compareInt(int a, int b) {
    //TODO: add code below
}
```

Note: You should use the if statement to compare a and b and use System.out.println to print the comparison results.

2. TEST: Add code in the main() class to test the above method:

- Prompt and use the Scanner to read two numbers a and b from Console
- Call the method compareInt and pass two numbers a and b

Lab Assignment

Similar to part 4, you are required to create a new method that receives two input parameters: user's salary and tax rate (both are double) and calculate the tax owed. The method should return the owed tax as double also.

1. Implement a new method in the class:

```
/**
 * This method calculates the owed tax of the user based on the input salary
 *   and tax rate
 * @param salary: the input user's salary
 * @param tax_rate: the tax rate apply to such a user (0 - 100%)
 * @return: the owed tax
 */
public static double calcTax(double salary, double tax_rate) {
    //TODO: add code below
}
```

2. Modify the code of the **main**() method to test calcTax() method as follows:

- Add the code to use the Scanner object to read the user's input of salary and tax rates as two new double variables
- Use if statement to validate the user's input: salary should be a positive number and tax rate is between 0% and 100%. In case any of two inputs is not appropriate, show an error message and quit the program.
- In case the two inputs are appropriate, call the method calcTax to get the returned owed tax and display it to the console.

Lab Result Submission:

You need to submit the results of the following sections to the D2L assignment item of Lab#1:

- Complete the required Lab Assignment above

- Compress the whole Eclipse source code of your Lab01 project in zip format using the Export function