Lab 2 – Variable Definitions, Expressions, and Debugging

The purpose of this lab is to get used to documenting your code, see the distinction between variables and constants, and to experiment with java types, mathematical expressions, and operations.

You will be graded on your ability to follow directions. Remember to do all the steps IN ORDER.

# Part 1 Tasks

1. Create a working directory folder in your computer, named **Lab02**. Use this as the location you will work in for this lab.
2. You must use Notepad++/Sublime and the command line prompt/terminal for this lab. When using the command line prompt/terminal, make sure that you navigate to the **Lab02** directory so that you can run the java files.
   * Windows Users:
     1. Use the dos command, **cd**, to change folders, and navigate to the directory that contains your source code file. (You can find a handy listing of some basic dos commands at this site, <http://www.cs.unca.edu/~jkdawg/help/msdos.html>[.)](http://www.cs.unca.edu/~jkdawg/help/msdos.html.))
   * Mac Users:
     1. Use the bash command, **cd**, to change folders, and navigate to the directory that contains your source code file. You can find a handy introduction to some basic bash commands at this site: https://[www.makeuseof.com/tag/mac-terminal-](https://www.makeuseof.com/tag/mac-terminal-commands-cheat-sheet/) commands-cheat-sheet/
3. Open up Notepad++/Sublime and create a new file called **VariableDefinitions.java**. Save the file in the **Lab02** directory. This is the Java source code file for your first program, so capitalization is important – the first letter must be capitalized and all the rest should be lowercase.
4. Inside the main method of **VariableDefinitions.java**, complete each of the following steps:
   1. Declare and initialize four of the variables in Part A of the PreLab Assignment. Make sure to have a comment line as shown below over each variable declaration. Save and compile your program, correcting any compilation errors before continuing each step. An example of this is shown below, but remember you need FOUR different variables.  
        
      // initial declaration of numPeople variable

byte numPeople = 3;

* 1. Add statements to print out the name of **each** variable and its value from Step a. Compile and execute your program. Again, an example of this is shown below but remember you need to print the four variables provided in part a.  
       
     System.out.println("The numPeople variable has a value of " + numPeople);
  2. THEN, write code to modify the values of the variables from Step a with a new assignment statement. Make sure you comment it as shown below. Compile and run your program. Again, an example of this is shown below but remember you need to modify the four variables from part a. Note, these modifications should be AFTER the print statements, do not modify the original combined declaration and initialization.  
       
     // modified value of the numPeople variable

numPeople = 6;

* 1. Add code to print out the names of the variables and their **NEW** values a second time, compile and run your program. This should be an additional four print statements. An example of one of them is below:  
       
     System.out.println("The numPeople variable now has a value of " + numPeople);
  2. NEXT, write statements to declare and initialize four constants, each one from a different primitive type in Java. Make sure you comment it as shown below. An example of this is shown below, but remember you need FOUR different variables.  
       
     // initial declaration of count constant

final int COUNT = 42;

* 1. Add statements to print the name of the constant and its value to your VariableDefinitions.java program and compile the program. Again, an example of this is shown below but remember you need to print the four variables provided in part e.  
       
     System.out.println("The COUNT constant has a value of " + COUNT);
  2. Add an assignment statement that will try to change one of the constant values. What happens?
  3. **Comment out the lines of code that cause compilation errors** but do not delete them and recompile your program.

# Part 2 Tasks

1. In the assignment, click the link to the **Lab 02 - Test Files.gfolder** and download to your **Lab02** directory the files linked inside **Lab2Files**. You will see ten java files sequentially named Test1 through Test10. Make sure that you save them with their respective filenames.java (Test1.java).
2. Compile and execute each of these files. Many of the files have errors that you will need to correct before running the program. **As you are debugging, you may not rename any of the file names**. You are to document any errors that you find in the quiz provided in this lab; correct the error; and execute the program.
3. Complete the Lab 02 Part 2 Quiz on the errors you find. You will have unlimited attempts on this quiz.

# Part 3 Tasks

1. Upload **VariableDefinitions.java** to gradescope

# Rubric

| Topic | Points |
| --- | --- |
| VariableDefinitions.java compiles | 8 |
| Lab Part 1a Variable declaration (1pts/declaration, 1pt per comment) | 7 |
| Lab Part 1b Printing appropriate variables (0.5pt per) | 7 |
| Lab Part 1c Modify (1pts/assignment, 1pt per comment) | 7 |
| Lab Part 1d Re-Printing (0.5pt per) | 7 |
| Lab Part 1e Constants Declaration (1pts/declaration, 1pt per comment) | 7 |
| Lab Part 1f Constants Printing (4 parts) (0.5pt per) | 7 |
| Lab 02 Part 2 Quiz | 50 |
| Total | 100 |