بسم الله الرحمن الرحيم

EE 367 : Data Structures and Algorithms
Faculty of Engineering
King Abdulaziz University
Department of Electrical and Computer Engineering

LAB #1

Compiling and Running Java Programs

Objective

This lab will have you practice editing and running Java programs using your favorite IDE, and accessing online Java documentation. It also shows you three error types.

Note: Text in **Green** color is for the attention of the TA.
Text in **Red** color is a shortcut for the student's required work.

There are several things you need to have your TA check off so you can get credit for this lab. You can have them checked off all at once when you are finished, but don't wait until the last minute--another lab may start as soon as yours ends, so ask to be checked off when there is still plenty of time. Checkoffs will not be done outside of your lab time.

PART I: Compile-time (Syntax) errors (2 points)

1. Start by getting the file called **`Names.java`** from the course web site ---> Lab Worksheets ---> Lab1.
2. Save the file **`Names.java`** in the workspace defined for your IDE.
3. Start your IDE (e.g. *Eclipse*), and open the file **`Names.java`**. This is a simple Java program for performing various string operations on a name. It almost works, but you need to make some changes so that it compiles and runs correctly.
4. Compile the program within your IDE. If successful, this will produce a java byte code (**`.class`**) file that can be used to run the program.
5. You will find that the program can't be compiled. The code contains two *syntax errors* which you are quite likely to make when you will write programs. (The second syntax error won't appear until you fix the first one.)
6. Use your IDE's **build** output facility to figure out what's wrong, correct the errors, save the file, and compile the code.
7. When you succeed to compile the program, take a snapshot of that and save it as ***part1***.

PART II: Run-time error (2 points)

1. Once you are able to compile the code, a file called **`Names.class`** will be created in the same directory.
2. Run this program within your IDE. The program will produce an error.
   How can you tell there is something wrong? This type of error, which occurs at run-time, tends to be significantly more difficult to correct than compile-time errors. (It's still somewhat easier than discovering an error in which the program appears to finish without problems, but is doing some computation incorrectly.)

3. Try to figure out what is the meaning of the error message. It may be hard to read at first, but it will allow you to answer certain questions, like: What is the method (i.e., function) that generated an error? What is its line number within the file **`Names.java`**?
4. The error is in one of the methods in the String class, which is a standard Java library. Your textbook may contain some documentation of the Java library, but the best source is the online documentation.
5. Find the documentation of the string function in error from the Java API documentation pages on the Internet. You can find the link in our course web site. There are two libraries that you will be using early in the semester—the java.lang package and the java.io package. Documentation on String and other standard data types is found in java.lang, so go there and find the String class. There is a lot of information, not all of which will make sense right now, but you should be able to find a description of the problematic String method.
6. Report how you found this information and how you figured out from it what was wrong with the program. Save this report as ***part2***.
7. When you think you have found the error, correct it, save the file, recompile, and execute it to see if the problem is solved.

PART III: Logical error (2 points)

1. Upon running your program successfully, Note the produced output. It has one single error in the result! The program will not produce any error messages! This type of error, is the most difficult to discover, since the program appears to finish without problems, but is doing some computation incorrectly due to error in its logic.
2. Read the output carefully, to discover where and what the error is. Once the error is discovered, report it as ***part3***.
3. Logical program-errors can often be discovered by monitoring the suspected variable(s) as the program advances through execution and comparing their actual values against what they ought to be. Use your IDE debugging tools to monitor and discover the cause of the error. Once variables in question are singled out, report what you did to find them as ***part4***.
4. Do any necessary changes to your program, so it produces the correct output. Report where and what changes you did as ***part5***.

Aside: You may think that the file produced by the Java compiler is named **`Names.class`** because the input file is named **`Names.java`**. Not so--the name of the .class file is based on the class name in **`Names.java`**. To experiment with this, change the line "**`class Names {`**" to "**`class Test {`**" and recompile.

Check-off

When you're done, assemble all five parts into one .pdf file and submit to your TA. Explain how you accessed the Java String documentation and found the bug. Also explain how you used the IDE debugging tools to find the logical error bug.

2 points: For showing that your **`Names`** program works.
2 points: For briefly explaining how you discovered and fixed the run-time error bugs, and showing how you accessed the Java String documentation.
2 points: For briefly explaining how you discovered and fixed the logical error bugs, and showing how you used your IDE debugging tools.