

APS 105S – Computer Fundamentals

Assignment #2: Basic Java

Winter 1999

(To be completed in your lab period January 18/19)

Objective

Learn how to use basic features of the Java Development Kit, and how to write simple programs that perform input, calculate results, and perform output, using variables, arithmetic operators, branches and loops.

1 Java Command Line Compilation

In this part of the lab you will learn how to compile and run a Java program using the Java Development Kit (JDK). Unless you have already done so, create a directory called `aps105` in your home directory. Make `aps105` your working directory. If you've forgotten how to do these things, look at lab #1 again. Use the following command to make a copy of the Java program `Hello.java` which is in `/share/copy/aps105`:

```
sparkl2.ecg% cp /share/copy/aps105/Hello.java .
```

If you use the `ls` command you should see a copy of the file sitting in the directory. Look at the file using a text editor. The program writes a simple message, "Hello world!". You can change the message to something more interesting if you wish. If you do, don't forget to save your changes.

When you write a Java program it is important that:

1. the filename is the same as the class name (in this case, it is `Hello`), and
2. the file ends with the `.java` extension.

You are now ready to compile and execute the `Hello` program. Enter the following command.

```
sparkl2.ecg% javac Hello.java
```

Use `ls` to list the directory contents. You should see a new file `Hello.class`. This is the compiled version of your program created by `javac`, the Java compiler. The program can now be executed (run) by entering the following command:

1

```
sparkl2.ecg% java Hello
```

The message should be output on the screen.

Now copy the file `Average.java` to your working directory from `/share/copy/aps105`. This file is a Java program that takes letter grades and calculates the approximate numeric average of the grades. Have a look at the program using the text editor, and then try compiling it.

You will notice that the program does not compile. Instead, several lines of error messages are printed. There are several small mistakes in the program, and the compiler has printed the messages to help you find the mistakes. Using the text editor, find and correct the mistakes. Remember to save the file after correcting the mistakes. Try compiling and running the program again. Try to find and fix all the errors before continuing with this lab.

2 A Simple Java Program With Input and Output

In the lectures several examples of simple Java programs were presented and discussed. Based on this discussion and on the examples above, write a Java program that finds the roots of a quadratic equation

$$ax^2 + bx + c = 0.$$

Obtain the values for the floating-point variables a , b , and c from the user and call the program `Quad.java`. The textbook has some additional information about the quadratic formula used to solve this equation on pages 64–6.

Your program should work (not stop with an error) for any reasonable values of a , b , and c . Your program should repeatedly request input for a , b , and c , and solve for the roots of the quadratic equation. When the input values are all zeroes, the program should end.

There are a number of "corner cases" that you must consider when writing your program. If the roots are complex, you must print out the complex conjugate pair. If there is only one root, then print it out only once. There are other "corner cases" you must find as well. **Predicting all of the "corner cases" that may arise and handling all of them gracefully is a very important aspect of good programming.**

Format the input and output as follows:

- Begin the program output with a title and center it.
- Give a brief description of what the program does.
- Prompt for each value on a separate line with a complete sentence, and have the user's input appear on the same line as the prompt. Use the routine `StdIn.getDouble()` to read the input from the keyboard; be sure to place a copy of `/share/copy/aps105/StdIn.class` in your working directory.
- Output the solutions on two separate lines.

The program should output something like what is shown below:

2

Quadratic Equation Solver

This program accepts three values from the user that correspond to the three variables in the quadratic equation:

$$ax^2 + bx + c = 0$$

and solves for the roots of the equation.

Enter values -- all zeroes will terminate the program:

Enter the value of a: 5

Enter the value of b: 6

Enter the value of c: -7

The solutions of $5x^2 + 6x - 7 = 0$ are

Root 1: 7.2665e-01

Root 2: -1.9266e+00

...function repeats until all zeros are input

Enter values -- all zeroes will terminate the program:

Enter the value of a: 0

Enter the value of b: 0

Enter the value of c: 0

End of Program

After you have this program working, you are ready to have your lab marked.

3 Optional Part

Sometimes it is nice to observe how the roots of a quadratic equation vary as one of the coefficients is varied. Modify your program to accept three values for the a coefficient: an initial value a_{ini} , a final value a_{final} , and an increment value $a_{increment}$. Have your program print out all the roots for the equation as a is varied from a_{ini} to a_{final} . To do this, start with $a = a_{ini}$ and print the roots. Then add $a_{increment}$ to a and print the roots again. Be sure to display the values of a used as you print the roots. Continue adding $a_{increment}$ to a and computing the new roots until $a \geq a_{final}$, then prompt for new input. Quit the program when all zeroes are entered. You can also try varying parameter b instead, or varying both a and b and displaying the roots in table form (difficult).