

## APS 105 – Computer Fundamentals

Assignment #3: Debugging, Loops and Conditionals

Fall 1999

(To be completed *before* your lab period, week of September 27)

### Objectives

Develop basic debugging skills and learn how to write a robust program, using if and while statements.

### 1 Debugging

When testing a program reveals a “bug” (something wrong with the program), you will need to be able to find the problem and correct it. In this part of the lab, you will debug an existing program in order to develop and practice your debugging skills. Once you have debugged the provided program, you may use it as a basis for the second part of the lab.

First, you must copy the program to be debugged, Average.java, from /share/copy/aps105/Average.java into your working directory. If you have not already done so, also place a copy of /share/copy/aps105/StdIn.class in your working directory.

Average.java contains several syntax errors, and so you will not be able to successfully compile it until you have fully debugged it. The only errors are syntactic, so once it successfully compiles you have completed this part of the lab.

### 2 Robust Java Program With Input and Output

In the lectures several examples of simple Java programs were presented and discussed. Based on this discussion and building upon the program you have just debugged, 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*. You can begin by making a copy of Average.java, but don't modify Average.java any more since you must be able to show both programs to the marker.

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.

1

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 roughly 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.
- Output each possible solution on a separate line.

The program should output something like what is shown below:

#### 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 zeroes 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 both programs working, you are ready to have your lab marked. Ensure that you have *thoroughly* tested and debugged Quad before having it marked so that it does not “crash” during testing.

2