AP Computer Science Java
Mr. Clausen
Program 6A

Program 6A LastNameFirstNameP6A (Quadratic Formula: 50 points)
(Including complex or irrational roots)

Write a program that begins by explaining the purpose of this program to the user. Don't forget to display a quadratic equation in standard form in this portion of the program ($ax^2 + bx + c = 0$). The purpose of this program is to allow the user to enter the coefficients a, b, and c, and calculate the solutions to the quadratic equation using the Quadratic Formula. The solutions will either be 2 real roots, one real root that is a double root, or 2 complex roots.

DO NOT USE a break statement to escape from any type of loop in this class regardless of what the book says or models.

1) Type comments at the beginning of the program to display your name and other information just like those used for program 2A.

2) Import the package:
   import java.util.Scanner;

3) Declare the class name in the format LastNameFirstNameP6A. Don't forget that the filename needs to be the same when you save your program. This would be a good time to save your program if you haven't done so already, LastNameFirstNameP6A.java.

4) Declare the main method:
   public static void main(String [] args)

5) There are no constants necessary for this program.

6) Instantiate an object of the Scanner class in order to use input from the keyboard. Use the statement:
   Scanner reader = new Scanner(System.in);

7) Declare all of the variables necessary for this program. You will need variables of type double to store: a, b, c, discriminant, realRoot1, realRoot2, doubleRoot, realPart, and imaginaryPart. Later you will need a variable of type String in order to ask if the user wishes to run the program again. Make sure that you use descriptive variable names to create readable, self-documenting code. (It's OK to use a, b, and c as variable names in this case, since this is a well known formula, but add comments to your variable declarations to make sure we know which is the coefficient of the quadratic term, linear term, and constant in the formula.)

8) Type the following comment:
//-------------------------Display My Information-------------------------
Follow this comment with println statements to display your name and period output just like those used for program 2A.

9) Type the following comment:
//-----------------------Display Program Explanation-----------------------
Follow this comment with println statements to display an explanation of the program. Don't forget to display a quadratic equation in standard form in this portion of the program ($ax^2 + bx + c = 0$).

10) For the Input section, type the following comment:
//-----------------------------------Input-----------------------------------
Ask the user to enter the coefficients of the quadratic equation, the values of a, b, and c, each in a separate input statement. Immediately after entering "a", use a "primed while loop" to "error trap" the value of the coefficient "a". Make sure that "a" is not zero. Give the user a message stating that if the value of "a" was zero, you wouldn't have a quadratic equation, but a linear equation. Be sure to ask the user to enter another value for the coefficient "a". After this loop, include an assert statement to guarantee that the value of "a" is not equal to zero.

11) For the Calculations section, type the following comment:
//---------------------------------Calculations---------------------------------
First, calculate the value of the discriminant ($b^2 - 4ac$). Next, use **extended** IF ELSE statements to calculate the solutions depending upon whether the discriminant ($b^2 - 4ac$) is a positive number (2 real solutions), is zero (one real solution multiplicity 2), or a negative number (2 complex roots). Remember, the quadratic formula is:

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

You will need to use the Math library for the absolute value (abs), power (pow), and square root (sqrt) methods. You do not need to import the Math package, it is accessible with using any import statements. For b^2 use the power method (pow) instead of multiplying "b" times itself. A summary of methods included in the Math class can be found in Appendix B, page B-5 in your textbook.

12) For the Output section, type the following comment:
//-----------------------------------Output-----------------------------------
For the output portion of your program, echo back the values of a, b, & c before displaying the solution(s). You will need extended if statements in this section of the program to determine which case of the solution to display. Tell the user whether there are 2 real roots, one real root multiplicity 2, or 2 complex roots in addition to displaying what the roots are. Remember, complex roots are binomials in the form a + bi. Also remember that complex roots occur in conjugate pairs. So if a + bi is one solution, then a - bi is a solution also. The "a" and "b" have nothing to do with the a, b, and c in the original quadratic equation. Remember also, the "a" in the complex root is a real number

(-1b)/(2a) and the "bi" is the imaginary number. You will have to be very careful when calculating this imaginary part and printing the "i" after it. Make this output easy to read and formatted nicely.

13) After you get this program working, add a "do…while" loop around the whole program asking the user if they would like to run the program again. If the answer is **not** "y" or "Y" then quit the program, otherwise run the program again. Since we are using a String variable to determine if the user wishes to run the program again, don't forget that we cannot check equality for strings with the = = symbol in our Boolean expression. You must use the "equals" method to test for equality in strings. ("do … while" is not part of the APCS Java Subset, but good to know. Information about "do … while" can be found in Appendix B, page B-4 in your textbook.)

When you run the program to test it, run it several times using the test data:

a=0 (just to make sure that your error trapping works)
a=1, b=0, and c= - 4
a=1, b= - 4 and c= 4
a=1, b= - 5 and c= 6 and
a= 1, b=1, and c=1

Don't forget to check your answers with paper and pencil to make sure they are correct. If your calculations are not correct, your grade will be reduced accordingly.

Use blank lines to separate each of the program sections listed in all of the steps above. Make sure that everything inside left and right curly brackets are indented three spaces from where the brackets are indented. Use good programming style as described in class. When you are finished with your program, have tested it thoroughly to make sure that your calculations are correct, and are sure that you don't need to make any changes, then save your program in the "W" network mapping, in the Program 6A folder. **Turn in the file named LastNameFirstNameP6A.java, I don't need the ".class" file for this program.**