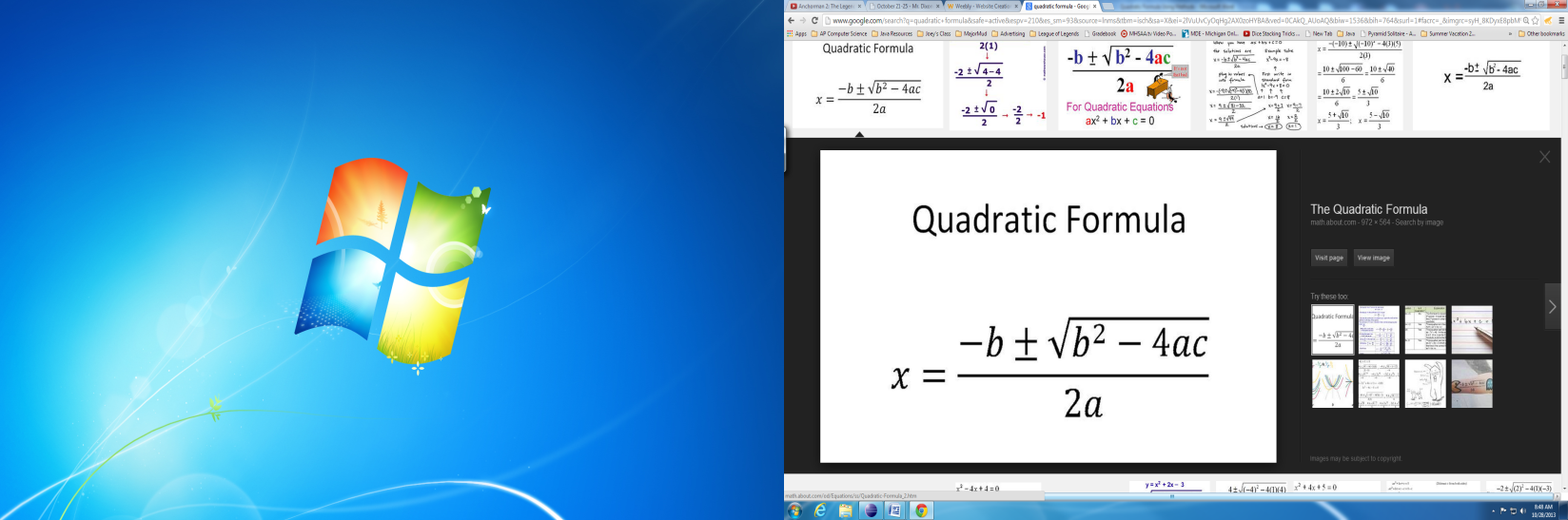
**Quadratic Formula Using Methods**

In this program you are attempting to solve the quadratic formula using a variety of methods to break up the calculation process. Your goal is to do this while declaring as few variables as possible in your main method. It is possible to do it with no variables declared but you will most likely want to declare a, b and c to keep your method calls easier to write.

Methods to be used:

Method # 1 - getValue() - This void method will use a Scanner to accept and return an integer value from the user.

Method # 2 - convertToDouble() - This method receives an integer value and returns its double equivalent.

Method # 3 - numberOfRoots() - This method receives a double value (discriminant) and returns an integer value of 0, 1 or 2, based on whether the discriminant's value is negative, zero or positive, respectively.

Method # 4 - calculateDiscriminant() - This method receives 3 double values and returns the value of the discriminant as a double value.

Method # 5 - isPositive() - This method receives a double value and returns true if its value is greater than zero.

Method # 6 - isNegative() - This method receives a double value and returns true if its value is less than zero.

Method # 7 - isZero() - This method receives a double value and returns true if its value is equal to zero.

Method # 8 - calculateSquareRoot() - Accepts a double and returns its square root. Used in outputRoot1, outputRoot2.

Method # 9 - squareIt() - Accepts a double and returns it squared. Must be used in the calculateDiscriminant method.

Method # 10 - outputTwoRoots() - Receives three doubles and calls outputRoot1 and outputRoot2 two roots (when numberOf Roots returns 2)

Method # 11 - outputOneRoot() - Receives three doubles and calls outputRoot1 ( when NumberOfRoots returns 1)

Method # 12 - noRoots() - Simply outputs that there are no roots for this equation.

Method # 13 - outputEquation() - Receives 3 doubles and outputs the formula in the format shown above.

Method # 14 - outputRoot1() - Outputs the root for -b +….. Must be used in outputTwoRoots, outputOneRoot methods. This method should receive a, b and c (three double values)

Method # 15 - outputRoot2() - Outputs the root for -b - ….. Must be used in the outputTwoRoots method. This method should receive a, b and c (three double values)

**public** **static** **void** main(String[] args)

{

System.***out***.println("Enter values for a, b and c");

//Get a, b, c from user

**double** a = *convertToDouble*(*getValue*());

**double** b = *convertToDouble*(*getValue*());

**double** c = *convertToDouble*(*getValue*());

**double** discriminant = *calculateDiscriminant*(a, b, c);

*outputEquation*(a, b, c); //output equation

//calculate and output answers based on the user input

**if**(*isPositive*(discriminant) || *numberOfRoots*(discriminant)==2)

*outputTwoRoots*(a, b, discriminant);

**else** **if** (*isZero*(discriminant) || *numberOfRoots*(discriminant)==1)

*outputOneRoot*(a, b, discriminant);

**else** **if** (*isNegative*(discriminant) || *numberOfRoots*(discriminant)==0)

*noRoots*();

}

**Sample Outputs**

Enter values of a, b and c: 10 14 -12

Solution for quadratic equation: 10x^2 + 14x - 12

x = 0.6

x = -2.0

Enter values for a, b and c: 1 2 3

Solution for quadratic equation: x^2 + 2x + 3

There are no roots for this equation.

Enter values for a, b and c: 1 2 1

Solution for quadratic equation: x^2 + 2x + 1

x = -1.0