The Quadratic Formula

A common problem in our math classes is to use the quadratic formula to find the roots of a quadratic equation of the form:

$$Ax^2 + Bx + C = 0$$

Here is the quadratic formula again in case you forgot:

$$x = \frac{-B \pm \sqrt{B^2 - 4AC}}{2A}$$

To take the square root of something in Java requires you to call a function called ***sqrt***. This function is in the Math library and to use it we need to put the word **Math** in front of the actual square root function. **This will be true of other math functions.** (As a note, we have already been doing this with printing stuff out, e.g. System.out.println(stuff)).

This function takes a **double** as an argument and returns a **double** as a result. Here is a thunk of code demonstrating this.

```
double resultvalue;
double numtotakeroot=42.0;

resultvalue = Math.sqrt(numtotakeroot);
```

You should search online for other math functions (hint, try "oracle java math class methods" in google).

Your task for this lab is to write a program that calculates the roots of a quadratic formula when the user inputs values for A, B, and C.

As you might recall, there are three kinds of results you can expect from solving a quadratic equation: two real roots, a double root and two imaginary roots. Your program should be able to:

1. Print out which of these three cases will occur for a given A, B, and C.
2. Print out the actual roots of the equation if the solution is not imaginary.
3. **Challenge:** Print out all roots, including the imaginary roots in standard form (a+bi).

Here are a few equations to test with:

Real Roots: $x^2 + 2x - 8$
Double Root: $16x^2 - 40x + 25$
Imaginary Roots: $2x^2 + 3x + 5$

The Pythagorean Theorem

Your task is to write a program that solves the Pythagorean Theorem. Ask the user to input two of the three sides of a right triangle and the program will output the missing side. Here is the formula if you have forgotten:

$$a^2 + b^2 = c^2$$

Some things to consider:

1. We need some way for the user to let the program know which of the variables to solve for. We can do this by having the user enter a specific value that the computer checks for before doing any calculations … in this case we can use -1 (since a triangle can't have a negative side length). Make sure to let the user know they need to input -1 to indicate the unknown side.
2. Solving for a, b, and c are all going to need their own equations.
3. **Challenge-** Have the program output the measures of the unknown angles in the triangle. (Hint: The math functions **asin(), acos(),** and **atan()** will find the inverse of the corresponding trig function.

New Stuff!

Ok. It's time to start making your own "methods", "procedures", "functions". I use all three names interchangeably because they are prevalent in the literature for all computer languages. Almost every object-oriented language (e.g. Java, C++, C#, Smalltalk, Object-C) use the word "method". The problem is that some methods acts as functions (you give them something and they give something back … like in math y= f(x) where we feed x to the function and it returns a value which gets put in y), and some methods just take info but return nothing. For example, your System.out.println(*string),* takes a *string* and prints something out to the screen, but it returns nothing to the program.

In this lab I want you to create to **methods** that simply do the jobs of the problems above.

One method will be called **quadratic** and one will be called **pythag.** Now, the cool thing about methods is that we can call them over and over wherever we need them. It won't matter in this program, but we will use that feature in future labs. In this program it will help the **readability** of the program. Here is how to structure the program:

class whatever etc.

public void quadratic() {

put here all the code that does the quadratic including variables etc.

}//quadratic

public void pythag() {

put here all the code that does the pythag etc.

}//pythag

public static void main(etc. etc.) {

put here your startup stuff where you ask the user which "problem" he wants to run..

}//main

}//original class name

As an example, suppose he told you he wants to run program 1, then your code would look like:

```
if (prognum == 1)
   quad();
else if  (prognum == 2)
   pythag();
```

etc. Notice how readable this is! It is very clear to the reader what is happening. We could have put all the code for "quad" right where the statement "quad();" is, but by packaging it up and sticking it somewhere else in the program, we have "encapsulated" its functionality.

The really cool thing is that we can call this over and over if we had to. We don't have that ability yet but it will be in the next lab. Notice also that "quad" and "pythag" are methods

that don’t return anything. That is what the word “void” means in the declaration. The word “public” means that we are making this available to anything in the program that may want to “call” (meaning ‘use’) it.