## INFSCI 0017 – Fundamentals of Object-Oriented Programming (Fall 2018)

## Lab 2

## Topics Reviewed

1. Variables
2. Constants
3. Data Types
4. Typecasting
5. Handling user input
6. Working with strings

## Grading and Submission

You are to write a complete Java program that meets the requirements outlined in the Lab 2 Tasks section.

Once you have completed the program, you should demonstrate your program for your Lab TA.

There will be 6 points for this lab, broken down in the following way:

1. Pythagorean Theorem Solution (2 point)
2. Area and Perimeter Solution (2 points)
3. Decomposing Money Solution (2 point)

Note that if your program does not compile, the TA will not grade it.

## Lab 2 Tasks

* Create a new Java project in Eclipse named YourPittID\_Assignment1. For example, your Pitt ID is abc123, your project should be named abc123\_Assignment1.
* Classes: there are 4 problems in this assignment, each of them has to be written in a different Java class (each in a different file). Remember name conventions for Java classes and class name and filename has to be EXACTLY the same.
* Do not use numeric literals – all numbers must be assigned to variables or constants! Any fixed factor used in a computation should be defined as constant.
* Follow the name conventions for naming your classes, variables and constants.  
  All the 4 problems can be solved with what has been covered in the first 2 weeks of class. Be aware that if you search for answers you might be given other solutions that may involve more advanced stuff. For this reason, we are giving precise enough instructions and tips.

**Problem 1: Solve Pythagorean theorem**

* Using JOptionPane, ask the user to input 2 numbers, corresponding to the sides of the triangle. Consider asking twice (one number each time) and storing them in String variables.
* Then convert those variables to numbers. Assume the user is kind and will enter a number (you do not have to check the input before converting it to number. We will see how to do this later on the course.)
* Create a double variable for computing the hypotenuse, and its value to the square root of the sum of the squares of the other two sides. For example if your variable for the hypotenuse is "c" and the sides are "a" and "b", then:
* Use the sqrt method of Java’s Math class to calculate the square root: Math.sqrt();
* Use the pow method of Java’s Math class to calculate the squares of the sides: Math.pow();
* Use JOptionPane.showMessageDialog() to display your result. In the output, give a complete sentence. For example "The hypotenuse is 5.0".   
  You might get an output with several decimal numbers and you may want to shorten the number to a certain amount of decimals. There are "advanced" ways to do that, but for now here is a simple trick: imagine you have the variable a = 15.327424302 and you want just to print 15.33 (rounding at the 2nd decimal).
* First multiply the number by 100 (a = a\*100). Now a will have the value 1532.7424302  
  Now round using Math.round (int r = (int) Math.round(a)). The value in r will be 1533.  
  Finally make a decimal division by 100 (a = r / 100.0). Now a will have the value 15.33.

**Problem 2: Area and perimeter of a circle**

* Using JOptionPane, ask the user to input the radius of a circle and convert the input to a double variable. In the example below, it is assumed the variable radius is named r. You can name it differently if you want.
* Compute the perimeter (2\*PI\*r) and the area (PI\*r\*r). Use the predefined constant Math.PI. You can use Math.pow to compute the r\*r if you want.
* Give an output in complete sentence ("The circle of a radius 10 has an area of 314.159 and a perimeter of 62.83"). If you have many decimals and you want to show less decimals when doing the output, see the trick at the end of the problem 1.

**Problem 3: Decomposing money**

* Ask the user to enter a number representing an amount of money from 1 dollar to 9999 dollars (integer). Assume the user is kind and will enter an integer number within this range.
* Output the number of "bucks" (single dollar), "sawbucks" (10s), "Benjamins" (100) and "grands" (1000) corresponding to the amount entered by the user. For example if the user enters 7528, the program should output:
  + 7 grands
  + 5 Benjamins
  + 2 sawbucks
  + 8 bucks
* TIP: you can decompose the total amount by using integer division and mod (%) operator. For example dividing by 1000 will give you the number of "grands", and then doing the mod 1000 will give you the reminder that has to be then divided by 100 to get the "Benjamins", and so on. You do not need to use loops.