TCSS142 Introduction to Object-Oriented Programming
Programming Assignment 3

DUE: Monday July 14, 2014 by 12 midnight

As would be expected, this assignment gives you experience with the use of methods that are passed values and produce a return result while using simple console input. Unlike the specific and singular purpose of previous assignments, 3's intent is more instructional and academic for the express purpose of acquiring a fundamental understanding of several concepts.

You will write a program that prompts the user for a radius of a circle, cylinder, and cone and a height of the cylinder and cone. You will read this information using the Scanner. The input information (radius and height) can be used to call several methods which will calculate: a line length which forms the hypotenuse in a cone based on the radius and height, the surface area and volume of a cone and cylinder, and the circumference and area of a circle. This information will be displayed in the console. This I/O (Input/Output) process will not be discussed until Wednesday's lecture.

The program will then prompt for the 3 dimensions of a box. This information will be sent to methods that calculate the surface area and volume of the box. As mentioned, most the return values will need to be displayed on the console. However, you will need to create your own formatting method that **receives a double value and returns a String** version of the double value that displays 2 decimal place accuracy rounded to the hundredth. You will also need a method that pads spaces, to the left of your output Strings, details are highlighted in red on page 2.

Finally, you will prompt the user to enter their first and last name (e.g. Scarlett Johansson) and using a Scanner object, read the **"entire"** name into a String variable (because of the space between names, you will need to **read an entire line, i.e. name = console.nextLine()**). Your program should then display the name in two formats (each should be done in separate methods that return the result strings):

1. All uppercase, Last name first, a comma, first name, e.g. JOHANSSON, SCARLETT
2. Original name string backwards: nossnahoJ ttelracS

Details of the above description are listed on page 2.

As the book points out, there are better means for input/output but, are much more cryptic to understand at an introductory level. Unfortunately, when a Scanner calls nextLine() (e.g. name = console.nextLine()) following a call to nextInt or nextDouble to read an int or double, a problem will occur. A newline character, left over from one of these previous calls, will be consumed by the nextLine() and thus, an empty string will be returned (Don't worry if you're not clear on this explanation. You will understand this issue later on). We won't concern ourselves over these small details. When your program finishes reading your geometric dimension values, insert the following line of code to consume the newline character just before you try to read in a person's full name:

```
console.nextLine();  // Consume the newline character left over from the last nextInt call
```

This seems wasteful but, is needed to prepare for a new line of input to be read by nextLine()..

Like all programs, method's purpose, and now, input and output should be clearly described before each method. Complicated code also needs documentation. Proper use of indentation and braces needs to be consistently implemented.

continued ---------- >

Your final program should include a brief javadoc statement to clarify each method such as:

```
/**
 *  circleArea method recieves a single parameter as a radius and
 *  calculates and returns the area to the calling method.
 */
public static double circleArea (double theRadius) { ...
```

You should also include multiline non-javadoc documentation at the beginning to identify the course, file name, programming assignment number, the due date, and the instructor. This is followed by a line of space and then just before the class declaration, javadoc to give a brief description of what the program does, an author tag for your name and the current date as a javadoc version tag, such as:

```
/*
 *  Course:             TCSS142 – Introduction to Object-Oriented Programming Summer 2014
 *  File Name:          Assign3.java
 *  Assignment:         3
 *  Due Date:           July 14, 2014
 *  Instructor:         Mr. Schuessler
 */
import java.util.*;    //  Note this statement.  Discussed in Wednesday's class.
 /**
  * This program calculates dimensions of various shapes...
  *
  * @author your name
  * @version 2014 July 12
  */
```

As mentioned in lecture, you will need to use an import statement to access the Scanner and related classes. I will supply your opening code here to get you started (you supply documentation):

```
// non javadoc comments here
import java.util.*;
// javadoc comments here
public class Assign3 {
   public static void main(String[ ] args) {
      Scanner console = new Scanner(System.in);
      double radius = 0.0;
      double height = 0.0;
      double circArea = 0.0;       // Used to store the area of a circle
      .
      etc.
      //  A bit easier way to declare multiple variables of the same type, note the commas between each:
      double length = 0.0, width = 0.0, boxHeight = 0.0;

      // Normally you should prompt and read data into your programs through method calls from main.
      // For this assignment you can prompt, input, and output from main.
      System.out.println("Enter a radius and height separated with a space: ");
      radius = console.nextDouble();
      height = console.nextDouble();

      circArea = circleArea(r);    //  method to calculate the area of a circle with radius r

      System.out.println( "\t" + padLeft("Circle Area:\t", 27) + padLeft(twoDigit(circArea), 14));
      // Above, the method twoDigit receives a double and returns a String representation of the double
      // value received using 2 decimal places accuracy rounded to the nearest hundredth.  The method padLeft
      // receives a String and an integer that represents an entire field width and returns the received String
```

```
    // combined with leading spaces to give an entire string length equal to the integer value received as the
    // second argument in the method.  This will create a right justified field when displayed.  E.G. The call
    // above: padLeft("CircleArea:\t", 27) returns a String of 27 characters padded with 14 spaces to the left
    // of the String "CircleArea\t" (the \t is a single character).
    etc.
  }  //  end of main

  public static double circArea(double theRadius) {
    return Math.PI * theRadius * theRadius;
    }

  public static String twoDigit(double theNumber)   // You have to solve this one.  The textbook might help.

  public static String padLeft(String theString, int theLength)  // You have to solve this one
```

NOTE:

Because padLeft accepts a String as it's first argument and twoDigit returns a String, also, twoDigit accepts a double as it's argument, the following is possible: padLeft(twoDigit(circArea(r)), width) where width is an integer variable (instead of specifically a variable, any integer expression will work).

You will design all of your methods (all formulas listed below). Keep in mind that you should be careful here to reduce redundancy. You will notice that several of the formulas use the same calculation(s) for multiple shapes, e.g. the surface area of a cylinder calculates the area of a circle as part of it's formula or the surface area of a box uses the same formula for the area of the sides. [Whenever possible your methods should take advantage of methods that already perform part of the task. E.g. The surface area of a box requires 2 * (the sum of the areas of the 3 unique rectangles which make up the sides of the box) : 2 * (rectArea(length, width) + rectArea(length, height) + rectArea(width, height))].

Though each one is fairly simple to implement, there are several methods in this assignment. You should create at least the following methods:

```
circumference  // has 1 double argument which is a radius and returns the circumference of a circle
                // You will need to decide what arguments and return values the remaining methods should
                // have.
circleArea     // Area of a circle
lineLength     // A right triangle is formed in a cone using radius as the base and the height.  You
                // calculate the hypotenuse formed in a cone based on the base and height.
coneArea       // Surface area of a cone
coneVolume     // Volume of a cone
cylinderArea   // Surface area of a cylinder
cylinderVolume // Volume of a cylinder
rectArea       // Used several times in the surface area of a box calculation
boxArea        // Surface area of a box
boxVolume      // Volume of a box
twoDigits      // Returns a String representation of a received double with 2 digit decimal
                // accuracy rounded to hundredth
padLeft        // Returns the received String padded left with spaces for an entire string length based upon
                // the second argument integer value.
switchNames    // Converts a First_Name Last_Name String to LAST_NAME, FIRST_NAME (all upper case)
                // and returns the latter
reverseName    // Returns a received String in reverse order
```

Formulas:

lineLength $\sqrt{radius^2 + height^2}$ calculates the hypotenuse formed opposite from the radius and height of a cone.

circumference	$2\pi r$	r is the radius
circArea	πr^2	
coneArea	$\pi r^2 + \pi rs$	surface area of a cone, s is the line length (hypotenuse)
ConeVolume	$\frac{1}{3}\pi r^2 h$	cone volume,
CylinderArea	$2\pi r^2 + 2\pi rh$	r is radius, h is height
CylinderVolume	$\pi r^2 h$	
rectArea	area of a rectangle - you need to do this one	
boxArea	you do this one	
BoxVolume	you do this one	

Save your program as Assign3.java **...this implies the name of your Class. Don't forget!**

Grading:

Program Compiles, passes tests, output correct:	60%
You have implemented the padLeft and twoDigits methods:	20%
Documentation (text and proper/consistent indentation), Proper use of methods to reduce redundancy, etc:	20%
	100%

Sample Run:

```
For circular shapes, please enter a radius and height separated by a space:
5 8

For rectangular/box shapes, please enter a length, width, and height separated by a space:
5 4 22

Circular shapes sizes using a radius of 5.0 and a height of 8.0:
             Circle Circumference:               31.42
                      Circle Area:               78.54
         Line length up cone side:                9.43
                Cone Surface Area:              226.73
                      Cone Volume:              209.44
            Cylinder Surface Area:              408.41
                  Cylinder Volume:              628.32

Box sizes using a length of 5.0, a width of 4.0, and a height of 22.0:
                 Box Surface Area:              436.00
                       Box Volume:              440.00

Please enter your First and Last Name: Mortimer Schnurd

Mortimer Schnurd in a LAST, FIRST format: SCHNURD, MORTIMER

Mortimer Schnurd backwards is drunhcS remitroM

Now isn't that silly!
```

Suggested Steps to Complete this Assignment:

1. It is best if you create a version of this program that only inputs the geometric shape information and generates the correct output without concerns for the formatting details, i.e. your padLeft and twoDigit methods (these can be added later once you have everything else working).

2. Add the input of a person's name (remember, you must read both names with a nextLine()) temporarily output the name to be sure it was read in correctly.

3. Implement the switchNames method

4. Implement the reverseName method

If you still have time:

3. Solve the twoDigit method and implement it in your code for testing.

4. Solve the padLeft method and implement it in your code for testing.

The majority of this program is straight forward but will contain a great deal of typed code. The last four methods listed on page 3 (twoDigits, padLeft, switchNames, reverseNames) will receive the majority of thought. For these reasons be sure to begin straight away and Good Luck!