CSc 110 Assignment 1: Introduction to Programming

How to hand it in:

Submit a file named **Welcome.java** through the Assignment 1 link on the CSC110 conneX site.

Learning Outcomes:

When you have completed this assignment, you should understand:

- How to design, compile, run and check a simple and complete Java program on your own.
- The effect of escape sequences on printed strings.
- How to write basic static methods.
- The flow of control (i.e. the effects of method calls and assignment statements).
- How to format and document a Java program.

Part 1 – Ascii Art

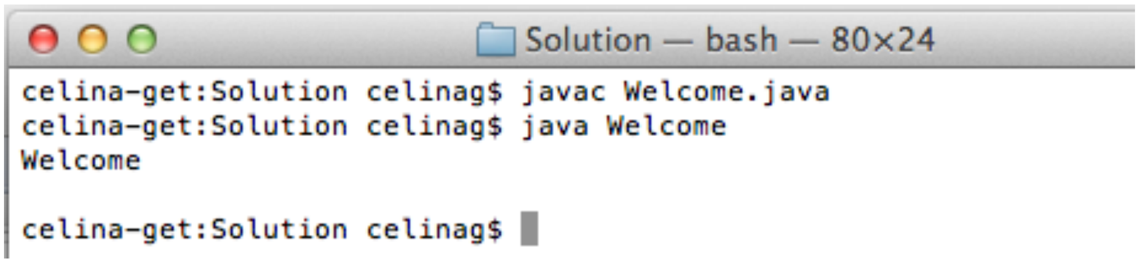
Write a set of methods that will output an ascii totem pole. Your implementation must include the three following static methods: `printTotemPole`, `printFrog` and `printPig`. You are free to introduce other methods if you would like but these three must be present and must be named EXACTLY as stated here.

Breaking down the steps

(see step 5 for what final output should look like)

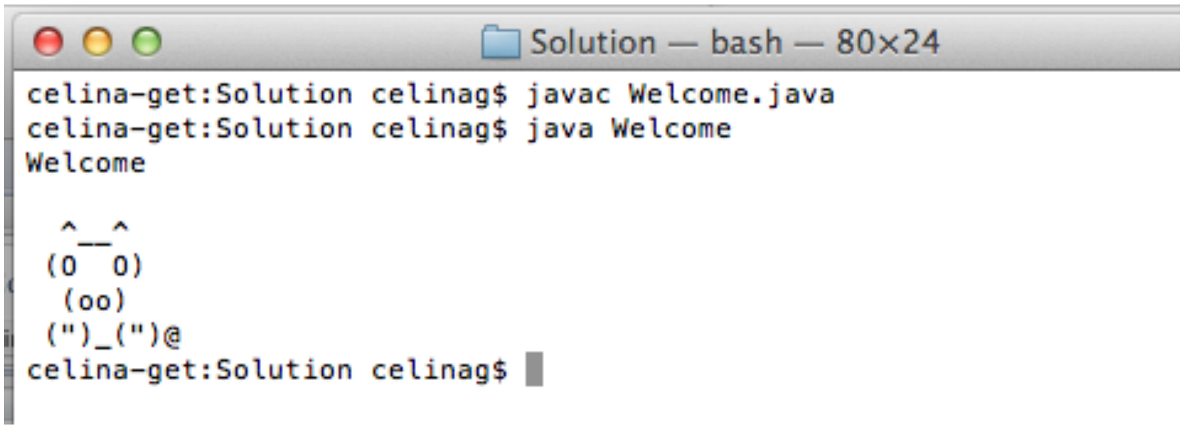
1. Create a public class Welcome
   a. Write the code for your class in your chosen editor
   b. Save this file as Welcome.java
   c. Compile this file in your terminal
      i. javac Welcome.java
   d. Repeat steps a through c until no errors are outputted to the console

2. Add a main method to your Welcome class in your editor
   a. Add the main method
   b. Add to your main method:
      a println statement that prints *Welcome* followed by an empty line to the console
   c. Compile in your terminal
      i. javac Welcome.java
   d. Run the program from your terminal
      i. java Welcome
   e. Repeat these steps until your output looks like the following…



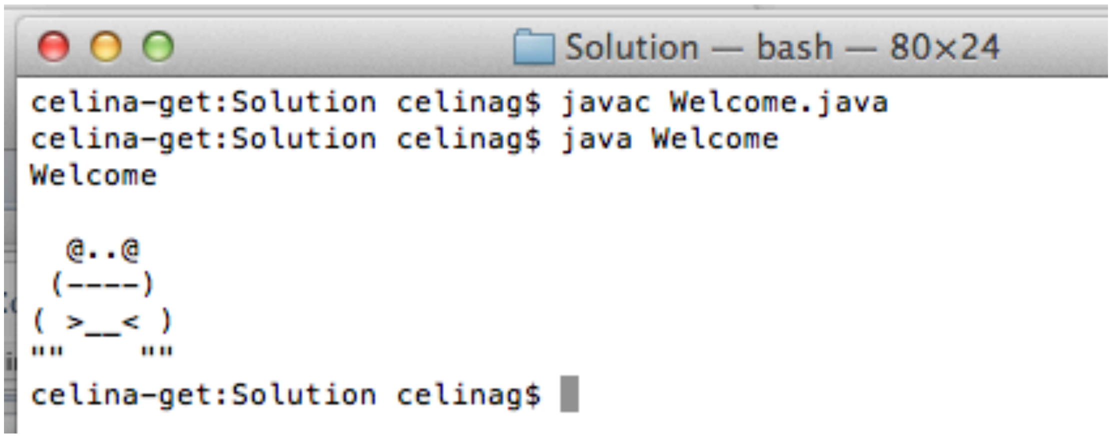
```
celina-get:Solution celinag$ javac Welcome.java
celina-get:Solution celinag$ java Welcome
Welcome

celina-get:Solution celinag$ 
```

3. Write and test the `printPig` method
    a. Write the code for this static method
    b. Call this static method from your main
    c. Compile your program
    d. Run your program
    e. Repeat these steps until your output looks like the following…
    f. Remove or comment out the call to this static method in your main method



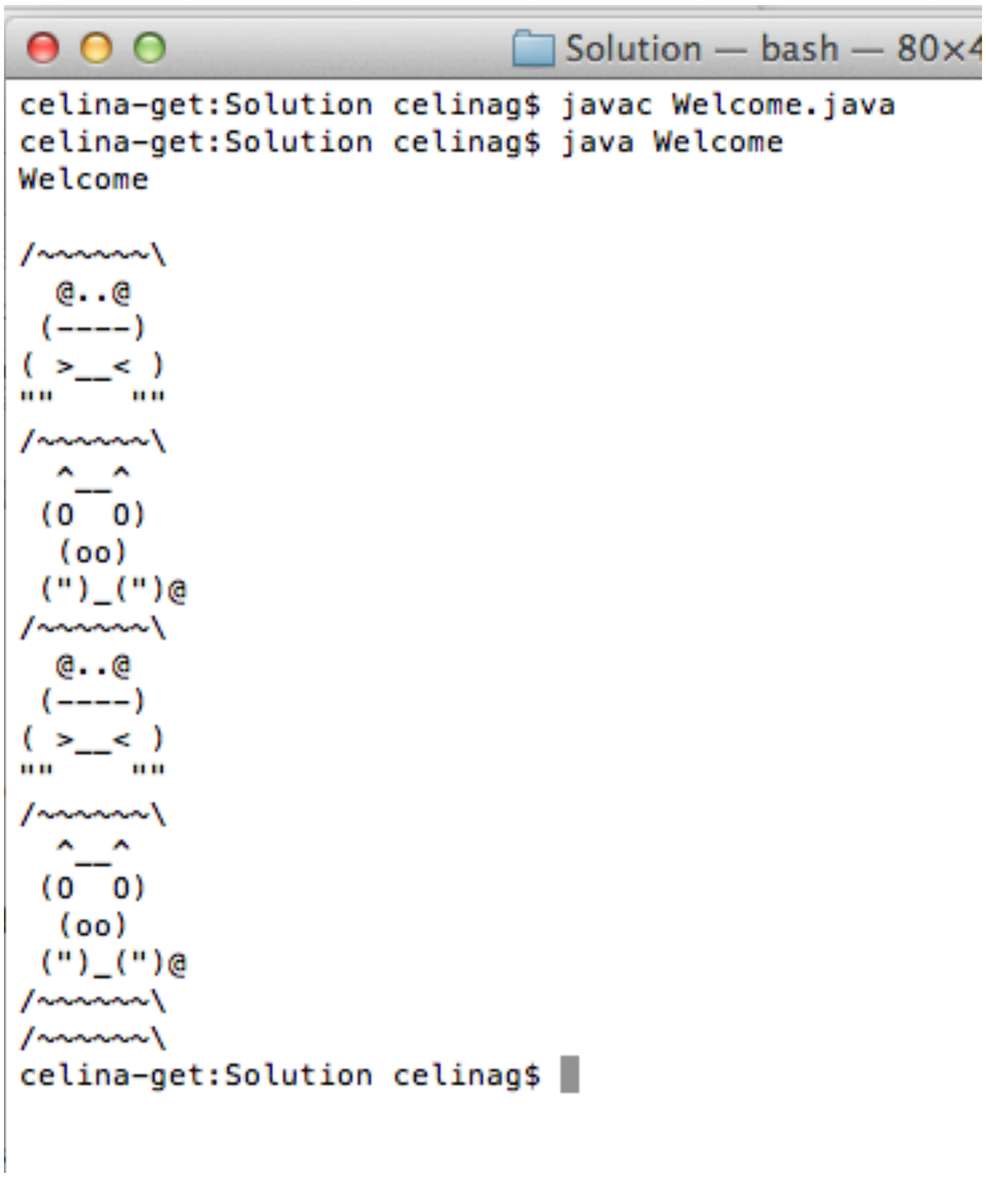
```
celina-get:Solution celinag$ javac Welcome.java
celina-get:Solution celinag$ java Welcome
Welcome

  ^__^
 (0  0)
  (oo)
 (")_(")@
celina-get:Solution celinag$ 
```

4. Write and test the `printFrog` method
    a. Write the code for this static method
    b. Call this static method from your main
    c. Compile your program
    d. Run your program
    e. Repeat these steps until your output looks like the following…
    f. Remove or comment out the call to this static method in your main method



```
celina-get:Solution celinag$ javac Welcome.java
celina-get:Solution celinag$ java Welcome
Welcome

  @..@
 (----)
( >__< )
""    ""

celina-get:Solution celinag$ 
```

5. Write and test the `printTotemPole` method
    a. Write the code for this static method
        i. You will be calling the previous two methods you wrote
        ii. Don't forget the spacer lines that are printed between each character!
    b. Call this static method from your main
    c. Compile your program
    d. Run your program
    e. Repeat these steps until your output looks like the following…

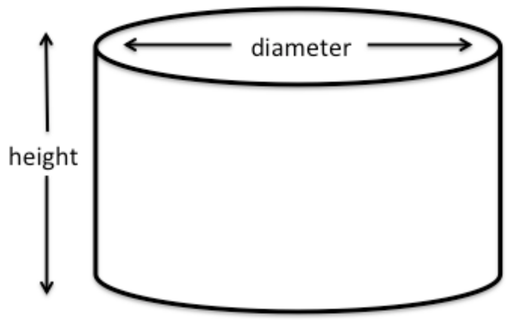


```
celina-get:Solution celinag$ javac Welcome.java
celina-get:Solution celinag$ java Welcome
Welcome

/~~~~~~\
  @..@
 (----)
( >__< )
""    ""

/~~~~~~\
  ^__^
 (0  0)
  (oo)
 (")_(")@
/~~~~~~\
  @..@
 (----)
( >__< )
""    ""

/~~~~~~\
  ^__^
 (0  0)
  (oo)
 (")_(")@
/~~~~~~\
/~~~~~~\
celina-get:Solution celinag$
```

NOTE: these ascii art figures looks best when your terminal uses a Courier font (ie. monospaced). Each character is composed of characters that include the: period, equal sign, double quote, forwardslash, backslash, ^, (,), an underscore and more!

Part II – Math Calculations

Write a method to calculate the surface area of a cylinder given the height of the cylinder is 5m and the diameter is 4m (diagram shown below).



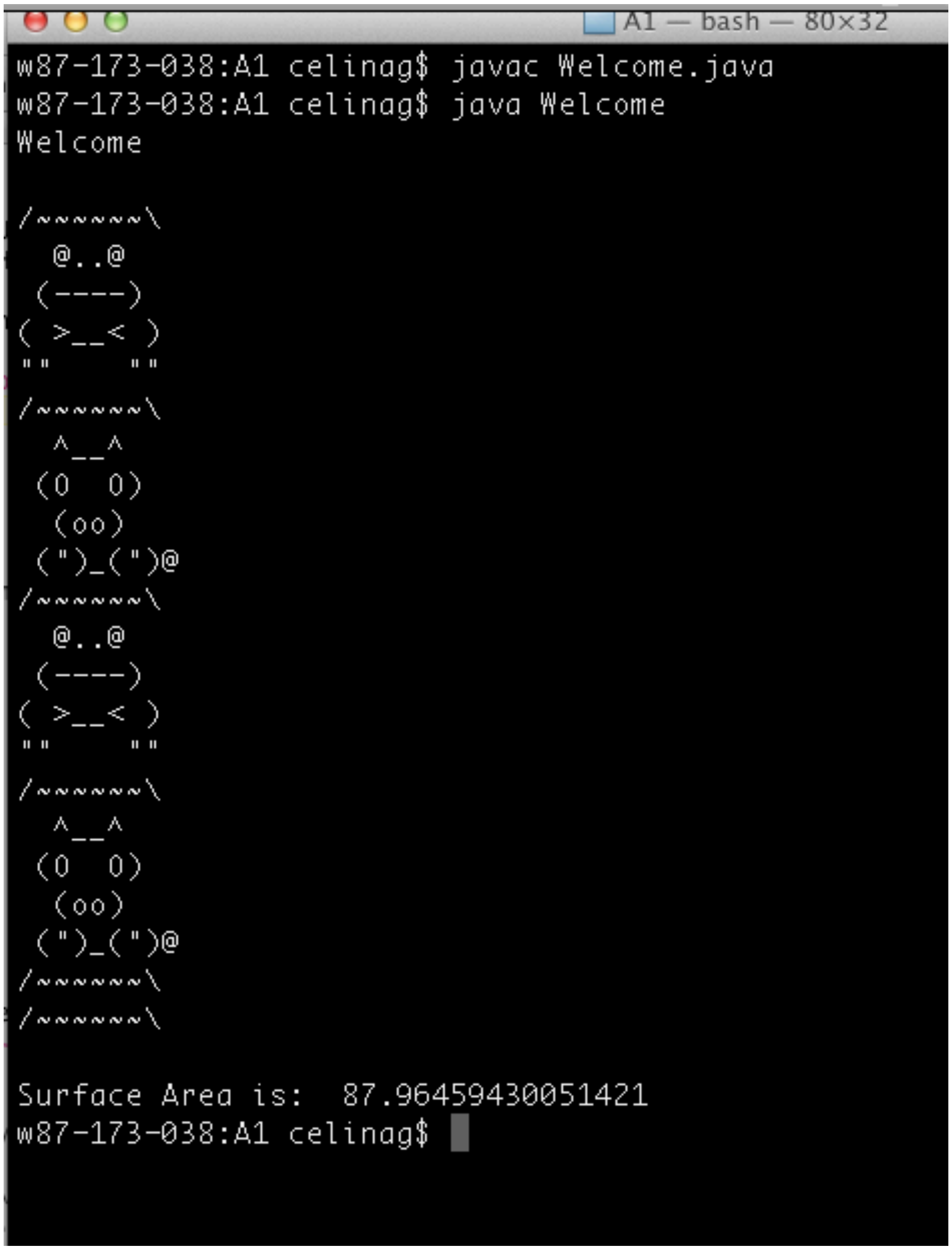
Ensure you adhere to the method specifications described below:

Breaking down the steps

1. Write a method called `calcSurfaceArea` that will calculate and print out the surface area of the cylinder to the console. Your method should use the following algorithm.
    a. Create 2 variables, one for `height` and set it to 5 and another for `diameter` and set it to 4.
    b. Using the `diameter` and the value of Π, write a statement to calculate the circumference of the cylinder and store the result to a variable that you create.
    c. Using the `diameter` and the value of Π, write a statement to calculate the area of the top of the cylinder and store the result to a variable.
    d. Using the `height` and the circumference you calculated earlier, write a statement to calculate the area of the walls of the cylinder.
    e. Using the 2 variables created above that are storing the area of the top and the area of the walls, write the lines of code to calculate the total surface area of the cylinder. Print the result to the console. Don't forget – the cylinder has a top and a bottom ☺
2. Call this method from your main after you print your totem pole.
   Additional Notes:

    - There are other algorithms to calculate the surface area of a cylinder but we are asking you to follow this algorithm to give you practice with the creation and use of variables.
    - chose a **primitive type** for your variables carefully in order to get a precise answer…
    - for the value of Π, you are free to: hard code this value, create a constant for PI, use the PI constant in java's Math library
    - chose the names of your variables carefully so they have meaning and make it easier for someone reading your code to understand what the code is doing
    - Challenge! Format your output so that only 2 decimal places are printed.

3. Write, compile and run your program until your output looks like that on the following page…



```
w87-173-038:A1 celinag$ javac Welcome.java
w87-173-038:A1 celinag$ java Welcome
Welcome

/~~~~~~~\
  @..@
 (----)
( >__< )
""    ""
/~~~~~~~\
  ^__^
 (0  0)
  (oo)
 (")_(")@
/~~~~~~~\
  @..@
 (----)
( >__< )
""    ""
/~~~~~~~\
  ^__^
 (0  0)
  (oo)
 (")_(")@
/~~~~~~~\
/~~~~~~~\

Surface Area is:  87.96459430051421
w87-173-038:A1 celinag$
```

Grading:

Marks will be allocated for the following…

- Your code must compile and run.
- Your code should produce the output exactly as requested.
- Your code must be indented and documented appropriately. Please follow the guidelines in `Style_Guidelines.pdf`, available in the Resources folder on conneX.