Java - Intro to Syntax/Output

Purpose

These labs were designed to introduce you to programming in Java by becoming familiar with syntax, semantics, compiling/executing a program and output to the console.

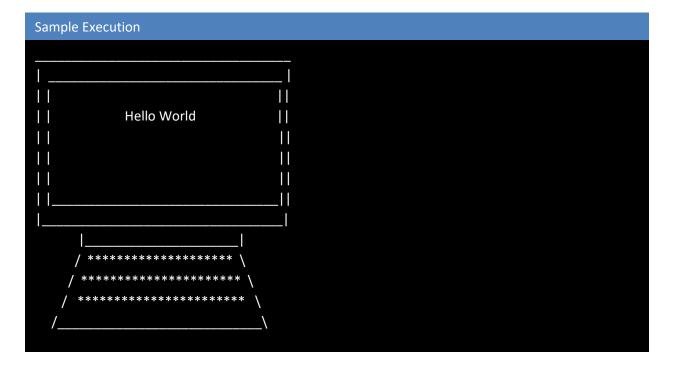
1.01 Welcome.java

Write a program in a class called Welcome.java that displays *Computer Science 1K, Welcome to Java,* and *Programming is Fun.*



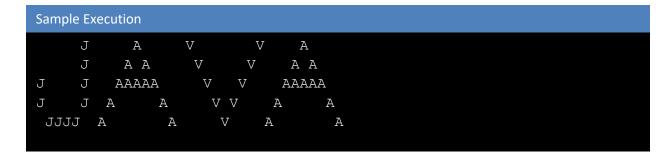
1.02 ASCIIArt.java

Write a program that displays a picture (animal, house, etc.) by outputting characters.



1.03 JavaPattern.java

Write a program that displays the word JAVA similar to the pattern below



1.04 Table.java

Write a program that displays an exponents table. "\t" or printf using %6s and %6d for Strings and integers may be useful.

```
Sample Execution
           x^2
                  x^3
      Х
      1
             1
                    1
      2
             4
                     8
      3
             9
                   27
            16
      4
                   64
            25
                  125
            36
                  216
```

1.05 HiBye.java

Write a program HiBye.java that takes two names as command-line arguments and prints hello and goodbye messages as shown below (with the names for the hello message in the same order as the command-line arguments and with the names for the goodbye message in reverse order). This can be run from the command prompt or finding out how to pass command-line arguments to the IDE. Either way you might need to google how to do this and how to access the args array from the main method.

```
Sample Execution

Hello Alan and Turing.
Goodbye Turing and Alan.
```

1.06 Reverse.java

Write a program that takes 4 command-line arguments and prints them in reverse order.

javac Reverse.java

java Reverse A B C D

Sample Execution

D C B A

1.07 AverageMPH.java

I ride my motorcycle 81 kilometers in 40 minutes and 20 seconds. Given that 1 mile is 1.6 kilometers, write a program that calculates the average speed in miles per hour. Be sure to use real numbers because Java performs truncation when dividing 2 integers. 7 / 4 mathematically is 1.75 but Java truncates or rounds down to 1. Alternatively, 7.0 / 4.0 is 1.75.

Sample Execution
75.3099173553719 mph

1.08 Pi.java

Write a program that calculates π . Use the following formula and stop when you reach 1/21.

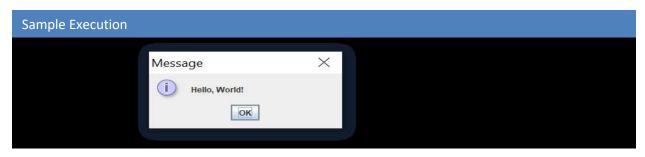
$$\pi = 4 * (1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \dots)$$

Sample Execution
3.232315809405594

1.09 Dialog.java

Write a program that says "Hello World!" using a graphical interface. Prior to your class add import javax.swing.JOptionPane; Inside of main add the line of code below.

JOptionPane.showMessageDialog(null, "Hello, World!");



1.10 TerminalColors.java

Write a program that outputs text in color to the terminal. Add the options below to your class(not inside main or you'll have to remove the *public static*).

```
public static final String ANSI RESET = "\u001B[0m";
public static final String ANSI BLACK = "\u001B[30m";
public static final String ANSI RED = "\u001B[31m";
public static final String ANSI GREEN = "\u001B[32m";
public static final String ANSI YELLOW = "\u001B[33m";
public static final String ANSI BLUE = "\u001B[34m";
public static final String ANSI PURPLE = "\u001B[35m";
public static final String ANSI CYAN = "\u001B[36m";
public static final String ANSI WHITE = "\u001B[37m";
public static final String ANSI BLACK BACKGROUND = "\u001B[40m";
public static final String ANSI RED BACKGROUND = "\u001B[41m";
public static final String ANSI GREEN BACKGROUND = "\u001B[42m";
public static final String ANSI YELLOW BACKGROUND = "\u001B[43m";
public static final String ANSI BLUE BACKGROUND = "\u001B[44m";
public static final String ANSI PURPLE BACKGROUND = "\u001B[45m";
public static final String ANSI CYAN BACKGROUND = "\u001B[46m";
public static final String ANSI WHITE BACKGROUND = "\u001B[47m";
```

System.out.println(ANSI_RED + "This text is red!" + ANSI_RESET);

System.out.println(ANSI_GREEN_BACKGROUND + "This text has a green background but default text!" + ANSI_RESET);

System.out.println(ANSI_YELLOW + "This text has yellow text but a default background!" + ANSI_RESET);

System.out.println(ANSI_CYAN_BACKGROUND + ANSI_RED + "This text has a cyan background and red text!" + ANSI_RESET);

```
This text is red!
This text has a green background but default text!
This text has yellow text but a default background!
This text has a cyan background and red text!
```