CSIS 10A Lab 3

Part A: Correcting Logic Errors and Math Formulas

Prerequisite: Lecture or Reading on Chapter 3 Sections 1-4

Please work with a partner.
When you are finished with both parts, call the instructor over for signoff
To begin, clone the git repository from https://github.com/mpc-csis/csis10a-lab-03 . Double click the package.bluej file in the lab03-bluej directory to open BlueJ on it.

Recall that a logic error is an error in the meaning of what your program is expressing. The computer follows the program instructions faithfully, but the answer is wrong because the instructions we write sometimes are expressed incorrectly. This happens frequently with math formulas.

First compile and run the `NumericTypes` program and observe the output.

1. Because of the comments, only Problem 1 will execute. It computes the average of two test scores. You should notice that the output is ***incorrect***. Please locate and **correct the logic errors** in the average formula. HINT: The logic errors could be due to conversion between data types, order of operations, or formula problems.

   The formula for averaging numbers is:

   $$average = \frac{score1 + score2}{numberOfScores}$$

   Make sure that the output makes sense before you continue. The average of 95 and 100 should be 97.5.

   To continue, remove the first of two slashes on the Problem 1 comment. This comments out the entire problem 1 code. Then add a slash on the Problem 2 comment. This activates the entire problem 2 code.

2. The temperature that water boils is 212 degrees Fahrenheit, equivalent to 100 degrees Celsius. In this problem we are trying to convert temperatures from Fahrenheit to Celsius. But the answer is not correct. Please locate and **correct the logic errors** in the formula. The formula to apply is

   $$C = \frac{5}{9}(F - 32)$$

   Make sure that the output makes sense before you continue. The temperature that water boils is 100 degrees Celsius.

   To continue, remove the first of two slashes on the Problem 2 comment. This comments out the entire problem 2 code. Then add a slash on the Problem 3 comment. This activates the entire problem 3 code.

3. Now you are going to write a program to compute the volume of a sphere given its diameter.

   a. A diameter variable is already declared and initialized.
   b. The diameter is twice as long as the radius, so calculate and store the radius in an appropriately named variable.
   c. The formula for the volume of a sphere is

$$V = \frac{4}{3}\pi r^3$$

   d. Add a line that calculates and stores the volume in an appropriately named variable. Use Math.PI for π and Math.pow to cube the radius.
   e. Print your results to the screen with an appropriate message.
   f. Compile, debug, and run using the following test data and record the results.

Diameter	**Volume (by hand)**	**Volume (by computer program)**

Part B: Void Methods

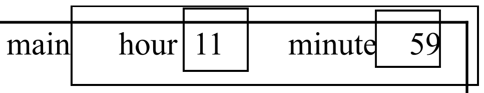
Prerequisite: Lecture or Reading on Chapter 3 Sections 5-10

Please work with a partner.
When you are finished with both parts, call the instructor over for signoff

Exercise 1 Here are two methods from a sample program. Draw a stack frame diagram that shows the state of the program when `main` invokes `printTime` with the arguments `11` and `59`.

```
public static void printTime(int hr, int min) {
    System.out.print(hr);
    System.out.print(":");
    System.out.println(min);
}

public static void main(String[] args) {
    int hour = 11;
    int minute = 59;
    printTime(hour, minute);
}
```



Exercise 2

The point of this exercise is to practice reading code and to make sure that you understand the flow of execution through a program with multiple methods. Here is a program:

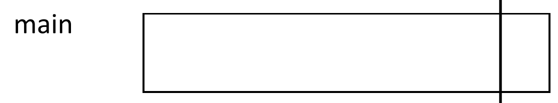
```java
public static void zoop() {
  baffle();
  System.out.print("You wugga ");
  baffle();
}

public static void main(String[] args) {
  System.out.print("No, I ");
  zoop();
  System.out.print("I ");
  baffle();
}

public static void baffle() {
  System.out.print("wug");
  ping();
}

public static void ping() {
  System.out.println(".");
}
```

Stack frame showing state of program after first time ping is invoked:



1. What is the output of the program ? Be precise about where there are spaces and where there are newlines.

   HINT: Start by describing in words what `ping` and `baffle` do when they are invoked.

2. Alongside the program above, please, draw a stack diagram that shows the state of the program the first time `ping` is invoked. Start with the frame for main and continue adding frames as each method is called, until the first time ping is called.

Exercise 3

The point of this exercise is to make sure you understand how to write and invoke methods that take parameters.

1. *Write the first line of a method named* `zool` *that takes three parameters: an* `int` *and two* `Strings`*. Right here:*

2. *Write a line of code that invokes `zool`, passing as arguments the value `11`, the name of your first pet, and the name of the street you grew up on.*
   Right here:

Exercise 4

The purpose of this exercise is to take code from a previous exercise and encapsulate it in a method that takes parameters. Inside the lab3 download, open the Date.java file. This is a student solution from Lab 1/2.

1. *Inside Data.java, write a method called `printAmerican` that takes the day, date, month and year as parameters and that prints them in American format. Once you set up the method header, you will basically move code from* `main` *and drop it into your* `printAmerican` *method.*
2. *Test your method by invoking it from `main` and passing appropriate arguments. The output should look something like this (except that the date might be different):*

   ```
   Saturday, July 16, 2011
   ```

3. *Once you have debugged `printAmerican`, write another method called `printEuropean` that prints the date in European format. Test this method by invoking it from* `main` *with the appropriate arguments.*