Lab 21 (November 20 or November 21)

Instructions: Complete the exercises below. Be sure to show your code to one of the lab TAs before you leave, so that you can receive credit for this lab. You must also upload a copy of all your source code (`.java`) files through the link on Blackboard by 11:59 PM on Tuesday, November 21.

1. Implement a Java method named `parseHex()` that converts a `String` (consisting of digits and uppercase letters ***ONLY***; for now, don't worry about lowercase letters or other characters) from hexadecimal (base 16) to decimal (base 10) and returns the result as an integer. Define a custom `HexFormatException` class that extends `Exception`. Your method should throw a `HexFormatException` if the string is ***NOT*** a valid hexadecimal value (e.g., if it contains any uppercase letters outside the range A–F).

   As a reminder, one basic (pseudocode) algorithm for translating from base N to base 10 is as follows:

   total ← 0
   currentPower ← 1

   *for each digit/position in the base N value, working from **right** to **left**:*
   *total ← total + (current digit's value * currentPower)*
   *currentPower ← currentPower * N*

2. This problem contains several steps:

   a. Design an interface named `Colorable` with a `public void` method named `howToColor()`. Every class of a colorable object must implement this interface.

   b. Design a new class named `Square` that extends `GeometricObject` (available from http://www.cs.armstrong.edu/liang/intro10e/html/GeometricObject.html) and implements the `Colorable` interface. Define the `howToColor()` method for a `Square` so that it prints a brief message on how to color the square (for now, this message can be anything you want, even something silly like "Use a blue crayon").

   c. Implement a class named `Triangle` that extends `GeometricObject` (but does ***NOT*** implement the `Colorable` interface). To speed things up, just provide *stubs* ("placeholder" methods used for testing purposes) for the two methods that every `GeometricObject` subclass must implement — `getArea()` and `getPerimeter()`. For now, each of these methods should always return 1.0.

d. Write a test program that creates an array of five `GeometricObjects` (a mixture of `Square` and `Triangle` objects). For each object in the array, if it is colorable, invoke its `howToColor()` method.

   Hint: Use Java's `instanceof` operator here to determine whether a `GeometricObject` is also a `Colorable` object. For example:

```
if (x instanceof Colorable)
{
  // Add Colorable-specific code here
}
```

Grading Guidelines: This lab is graded on a scale of 0-3 points, assigned as follows:

0 points: Student is absent or does not appear to have completed any work for the lab

1 point: Student has written only one program, but it does not compile or run at all due to errors.

2 points: Student has written (or attempted to write) both programs, but only one compiles and runs without error.

3 points: Student has written both programs, and they both compile and run correctly, without any apparent errors.