

Programming Exercise 12-1

In this assignment, start with a solution to Lab 4-2 (you may use your own solution or the one provided for you here) and improve the implementation by introducing proper exception handling.

In the lecture slides, we showed how to make use of an `IllegalTriangleException` whenever the 3 input sides do not form a triangle. Add this code to the current implementation.

Also, create a class `IllegalClosedCurveException`; modify `IllegalTriangleException` so that it is a subclass of `IllegalClosedCurveException`.

In each of the constructors and mutator functions of `Square`, `Rectangle`, `Circle` and `Triangle`, throw an `IllegalClosedCurveException` whenever any of the input dimensions (side, radius, etc) is not a positive number.

In the Test class, be sure to handle both kinds of exceptions; a catch clause should simply print to the console the type of exception and the name of the class in which it occurred, like this:

An `IllegalClosedCurveException` was thrown in a `Rectangle` instance
or
An `IllegalTriangleException` was thrown in a `Triangle` instance.

(For fun, optionally, you may indicate an error has occurred by using `JOptionPane` – this was done in the slides, but not necessary here.)