Lab 6B, Data Hiding

# Lab Intro & Prep

The exercises in this lab sheet will allow you an opportunity to practice using data hiding through access modifiers and using getters and setters to access such members.

## Learning Objectives

* Develop programs that utilise access modifiers to promote data hiding and encapsulation
* Become more familiar with an IDEs source-code generation (generating of getters and setters)

# Exercise 1 – Rectangle Class

Create a class Rectangle with **float** attributes **length** and **width**, each of which defaults to 1. Provide methods that calculate the rectangle's perimeter and area. It has set and get methods for both length and width. The set methods should verify that length and width are each floating-point numbers larger than 0.0 and less than or equal to 20.0. Write a program with a main method to test the Rectangle class to ensure the verification within the setter works.

Fig 1 shows an example session where two rectangle objects are created. The first rectangle has valid length and width values of 10 and 10, respectively. The second has a length of -35 set, meaning a message is output specifying it is invalid – meaning the default of 1 is kept. The second has a width of -6 – which is also invalid.

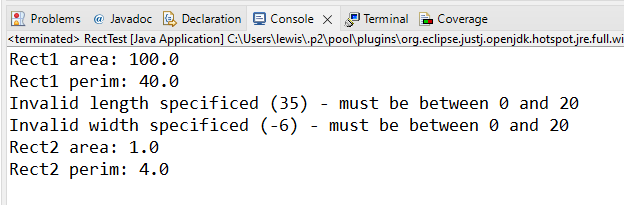


Fig 1 - Rectangle Program Output

# Exercise 2 – Time Class

Create a Time class that has three integers – hour, minute, and second. Do not provide a constructor (we can keep the default one which accepts no arguments). Provide a method to set the time (void setTime), in which the hour, minute, and second are taken as parameters. In the setTime method, verify that the inputs are within sensible ranges; if any of them are not, inform the user and print out the time that they tried to set.

Provide another method in the **Time** class named printTime. This method will print out the time in a string-friendly format.

Test the application by creating multiple Time objects and calling the setTime method with both valid and invalid hour/minute/seconds.

Fig 2 shows an example session. The first time object (time1), has the setTime method called, with the values 14 for hour, 25 for minute, and 55 for second. The printTime method is called to display the time. The second time object (time2), has the setTime method called with an invalid input (hour is 26), the printTime method is called, which shows that the setTime method has not taken effect.

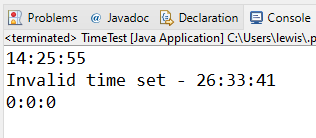


Fig 2 - Time Program Example Output

At this point, you may want to implement data hiding in your assignment in Processing. The concept works in the same way, regardless of the environment. You want to hide your classes' variables and provide methods to retrieve and set them (possibly checking that sensible values are set).