

Lab 05.1: Creating Classes

Objective

This lab will provide practice in creating classes, methods and attributes. You will also implement several small methods in your new class during this exercise.

Overview

In this lab you will:

- Create new classes, attributes and methods
- Implement your methods

Step by Step Instructions

Exercise 1: Creating Classes and Attributes

1. Create a new class named **Box** in the **com.lq.exercises** package.
2. A **Box** has three pieces of information
 - a. height
 - b. width
 - c. length
3. Create **private** attributes in your **Box** class for each of these pieces of information. Use `double` as the data type for each attribute.
4. Create **public** methods to set each attribute and retrieve (get) each attribute. Right click on the editor window and select the Source menu. From that menu select "Generate Getters and Setters". Check the "Select all" and "Generate Comments" boxes. From the location dropdown choose last member and then click finish.

Exercise 2: Create Constructors

5. Create two constructors for the **Box** class.
 - a. The first should accept 3 parameters and assign them to the **length**, **width** and **height** attributes respectively. Note: there is a wizard on the “Source” menu that will create this without typing.
 - b. The second should accept 1 parameter and assign it to all 3 attributes (creating a cube). This constructor should make a call to the other constructor to avoid duplication of code.

Exercise 3: Create Business Methods

6. Create a method named `getVolume` that accepts no parameters and returns a `double` containing the volume of the **Box**. [Hint: Volume is height * width * length.]
7. Create a method named `getSurfaceArea` that accepts no parameters and returns a `double` containing the surface area of the **Box**. [Hint: Surface area is the sum of the areas of the 6 sides of the **Box**.]

Exercise 4: Create a printBox Method

8. Create a method named `printBox` which does the following:
 - a. If any one of the 3 attributes is less than or equal to 0, it will print a message stating that the box contains invalid properties.
 - b. If all 3 sides are set correctly (i.e. greater than 0), it will print a message in the following form:

```
Length = 5.0
Width = 6.0
Height = 7.0
Volume = 210.0
Surface Area = 214.0
```

Note: we will not test any of our code at this point. Once we cover creating objects and calling class methods, we will ensure that all the code is operating correctly.

Challenge Exercises:

Refactor your code to eliminate duplicate logic.