CEGEP VANIER COLLEGE
CENTRE FOR CONTINUING EDUCATION
Advanced Programming in Java
420-984-VA

Teacher: Samir Chebbine **Lab 1** **Jun 27, 2022**

Lab 1: OOP and Inheritance

Complete all these following programs as explained in my **Lab 1 YouTube Video 1.** All *missing coding statements* are presented in this YouTube video with explanation.

Create and Submit a Word file ***Lab1OOPProgramminIIYourName.doc*** which contains Answers of Book Exercises and output screenshots for every Java Project. Submit the Java projects too.

1. Inheritance:

a) **Superclass or Base class vs. Subclass or derived class**

Create the Project **GeometryProject** of Figure 1. *Rectangle* class is called the *Superclass* and the *Box* class is called the *Subclass*. How many *Data members* declared in the Box class ?

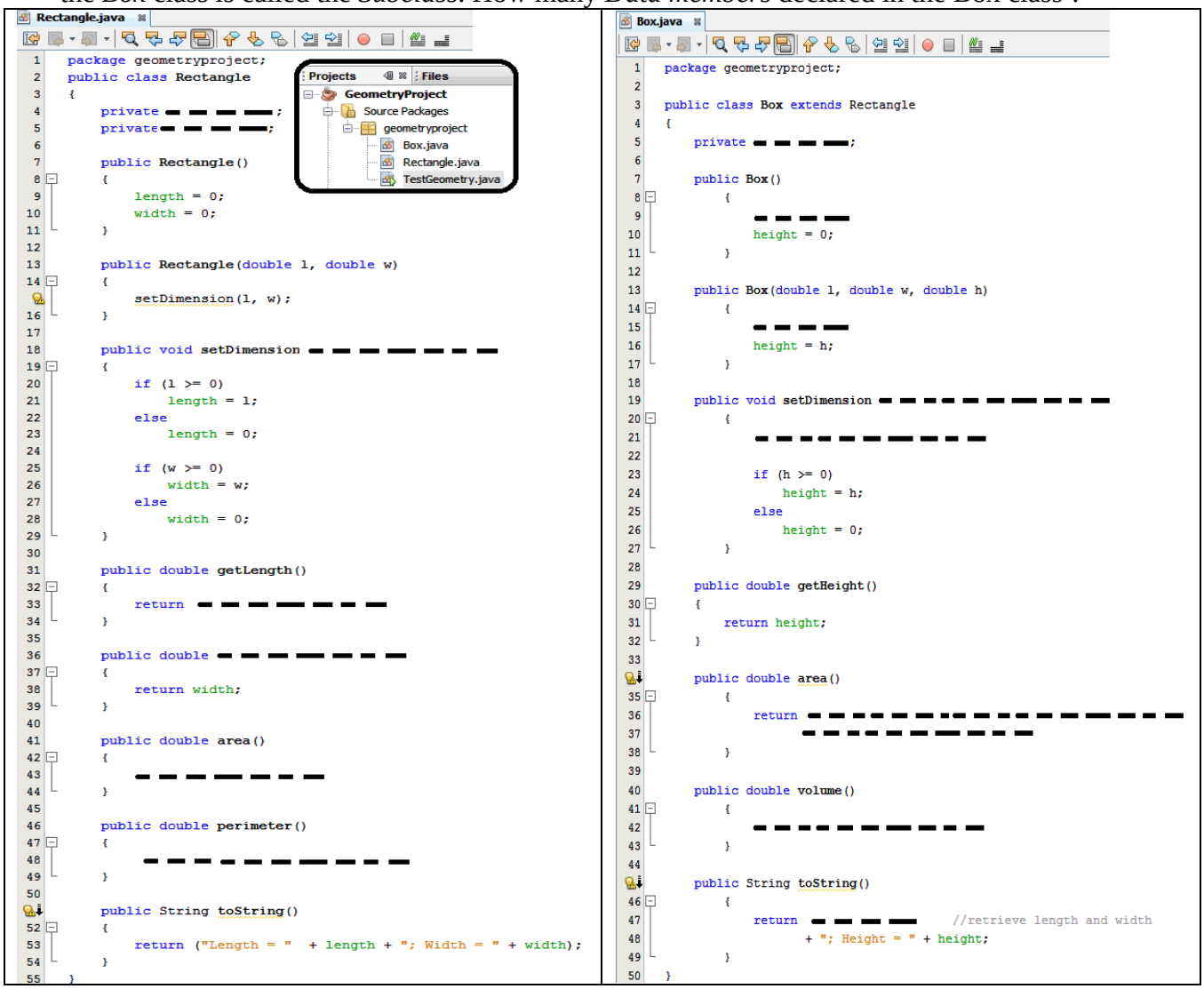

Figure 1

(Testing the *Rectangle* and *Box* Classes) Create the Java Program *TestGeometry.java* of Figure 2 to test if you can *instantiate objects* from *Rectangle and Box Classes* defined in Figure 1. How many objects have we created in *TestGeometry.java*? What is the purpose of using the operator *super* in Figure1?

TestGeometry.java

```
1  // This program illustrates how the objects of a superclass and a
2  // base class work.
3  package geometryproject;
4
5  public class TestGeometry                                    //Line 1
6  {                                                            //Line 2
7     public static void main(String[] args)                    //Line 3
8     {                                                         //Line 4
9        Rectangle myRectangle1 ▬ ▬ ▬ ▬▬ ▬ ▬ ▬                  //Line 5
10       Rectangle myRectangle2 ▬ ▬ ▬ ▬▬ ▬ ▬ ▬                  //Line 6
11
12       Box myBox1 = new Box();                                  //Line 7
13       Box myBox2 = new Box(10, 7, 3);                          //Line 8
14
15       System.out.println("Line 9: myRectangle1: "
16                ▬ ▬ ▬ ▬▬ ▬ ▬ ▬);                              //Line 9
17
18       System.out.println("Line 10: Area of myRectangle1: "
19                   ▬ ▬ ▬ ▬▬ ▬ ▬ ▬);                          //Line 10
20
21       System.out.println("Line 11: myRectangle2: "
22                ▬ ▬ ▬ ▬▬ ▬ ▬ ▬);                              //Line 11
23       System.out.println("Line 12: Area of myRectangle2: "
24                   ▬ ▬ ▬ ▬▬ ▬ ▬ ▬);                          //Line 12
25
26       System.out.println("Line 13: myBox1: " + ▬ ▬);         //Line 13
27
28       System.out.println("Line 14: Surface Area of myBox1: "
29                ▬ ▬ ▬ ▬▬ ▬ ▬ ▬);                              //Line 14
30       System.out.println("Line 15: Volume of myBox1: "
31                ▬ ▬ ▬ ▬▬ ▬ ▬ ▬);                              //Line 15
32
33       System.out.println("Line 16: myBox2: " + ▬ ▬);         //Line 16
34
35       System.out.println("Line 17: Surface Area of myBox2: "
36                ▬ ▬ ▬ ▬▬ ▬ ▬ ▬);                              //Line 17
37       System.out.println("Line 18: Volume of myBox2: "
38                ▬ ▬ ▬ ▬▬ ▬ ▬ ▬);                              //Line 18
39    }                                                           //Line 19
40 }                                                            //Line 20
```

Output - GeometryProject (run)

```
run:
Line 9: myRectangle1: Length = 0.0; Width = 0.0
Line 10: Area of myRectangle1: 0.0
Line 11: myRectangle2: Length = 8.0; Width = 6.0
Line 12: Area of myRectangle2: 48.0
Line 13: myBox1: Length = 0.0; Width = 0.0; Height = 0.0
Line 14: Surface Area of myBox1: 0.0
Line 15: Volume of myBox1: 0.0
Line 16: myBox2: Length = 10.0; Width = 7.0; Height = 3.0
Line 17: Surface Area of myBox2: 242.0
Line 18: Volume of myBox2: 210.0
BUILD SUCCESSFUL (total time: 1 second)
```

Figure 2

2. Using your own wording, answer the following questions briefly:

1. What is the purpose of OOP (Object Oriented Programming) and why use we Inheritance in Java?
2. What is the difference between key word ‘extends’ and ‘super’ used in Inheritance?
3. How many Data Attributes included in objects of Box class type?
4. How many Methods declared in the Box class?
5. Which object has more data attributes Rectangle class (Super class) or Box class (Sub class)?
6. Given *Employee* and *Person* as classes, which one is super class, which one is sub class?
7. Give an example of super class and sub class as **your own** project (different than Geometry and Employee/Person) and instantiate one object from super and sub classes.
8. True or false and why: In the definition of class template, which one is valid as per Inheritance?
   a. public class *Box* extends *Rectangle*
   b. public class *Rectangle* super *Box*

3. Complete the following project as explained in my **Lab 1 YouTube Video 2**.

Project SportProject: Create a Java Project ***SportProject*** using NetBeans IDE that allows end user to evaluate the cost of a given sport training.

a) You need to write a **Java class** called **`Sport`**, which takes Name, Number of hours per week, Number of weeks as three **private** `non static` members called respectively `name, number_hour, number_week`. A variable hourly rate of sport training `cost_hour` as **public** `static` data member. The `Sport` class should contain the following method members:
   1) Add **default constructor** (`name= "", number_hour=0.00, number_week=0`) and **constructor with parameters (get the same name as the name of the class)** within the `Sport` class in order to initialize the data members (`name, number_hour, number_week`) of every object.
   2) Add Accessor /Mutator methods to set the values of private members.
   3) Add a method called `CalculateCostTraining()` in `Sport` class to calculates and returns the cost of a training (`cost_training = cost_hour * number_hour * number_week`).
   4) Add a method called `public String toString()`in `Sport` class to print the Sport Training information in the form of `//name//number_hour// number_week//cost_hour$`

b) Write a program to be called **`TestSport`** that tests your class **`Sport`**. For this purpose:
   1) Instantiate two objects from **`Sport`** class type called `myPlayer`, and `yourPlayer`. Initialize `myPlayer` to (`"Bob", 3, 15`) and initialize `yourPlayer` to (`"Irena", 4, 25`) and return their respective cost of training `cost_training`, set the static member `cost_hour` to 15$.
   2) You should allow the user to input `name`, `number_hour`, `number_week` and `cost_hour`. Assign the entered values to the object `yourPlayer`. Return also its cost of training `cost_training` as shown in Figure 3.

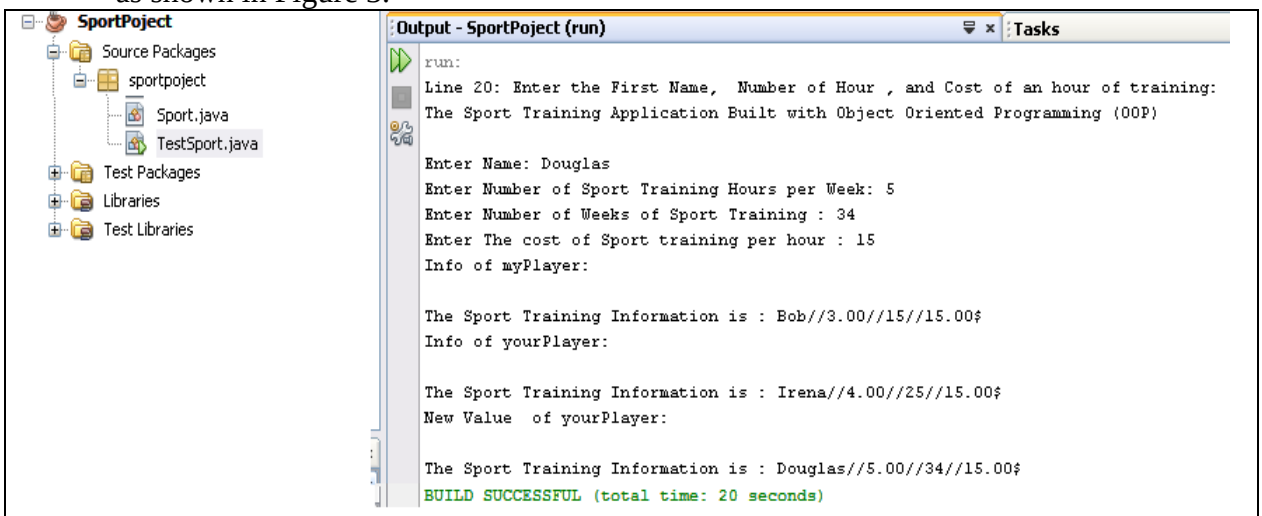

Figure 3

c) (**Inheritance**) Create a class, which is called **`OlympicSport`** that extends **`Sport`** and includes a private data member called *cost_pro* for professional sport training.

   1) Add toString() method for returning the values of data members, and an **overriding** method called *CalculateCostTraining()* that returns the cost of a given training:
   cost_training = (`number_hour * number_week * cost_hour`) + `cost_pro`.
   2) Within **`TestSport`**, instantiate two objects from **`OlympicSport`** class type called `olympicFaresPlayer (name, number_hour, number_week, cost_pro)`, and `olympicSarahPlayer`. Initialize `olympicFaresPlayer` to ("Fares", 2, 10, 6) and initialize `olympicSarahPlayer` to ("Sarah", 2, 18, 7). `cost_hour data attribute` is set previously from console to 15.

3) Display all information related to object `olympicFaresPlayer` and `olympicSarahPlayer` using the defined `toString()` method as shown in Figure 4.

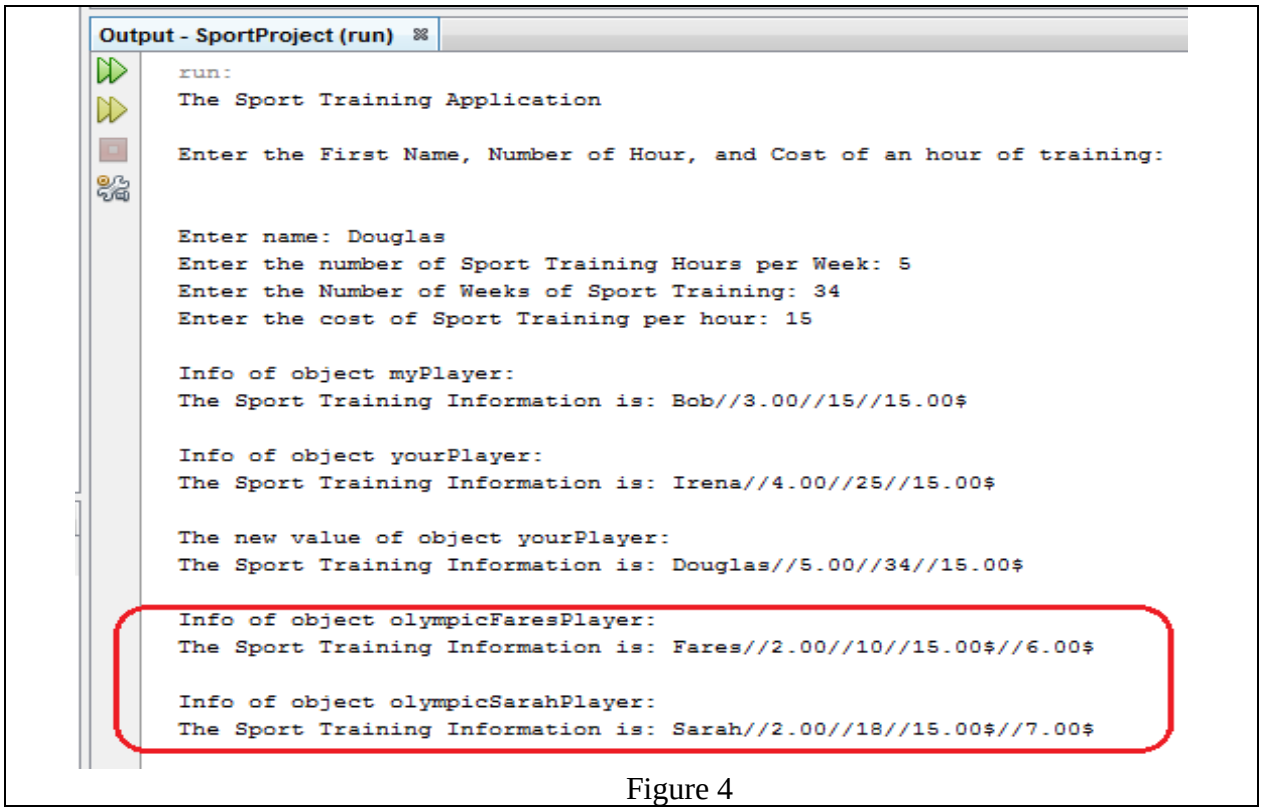

Figure 4

4) Display all information related to object `olympicFaresPlayer` and `olympicSarahPlayer` using the defined `getter(s)` and return their respective cost of training *cost_training* as shown in Figure 5.

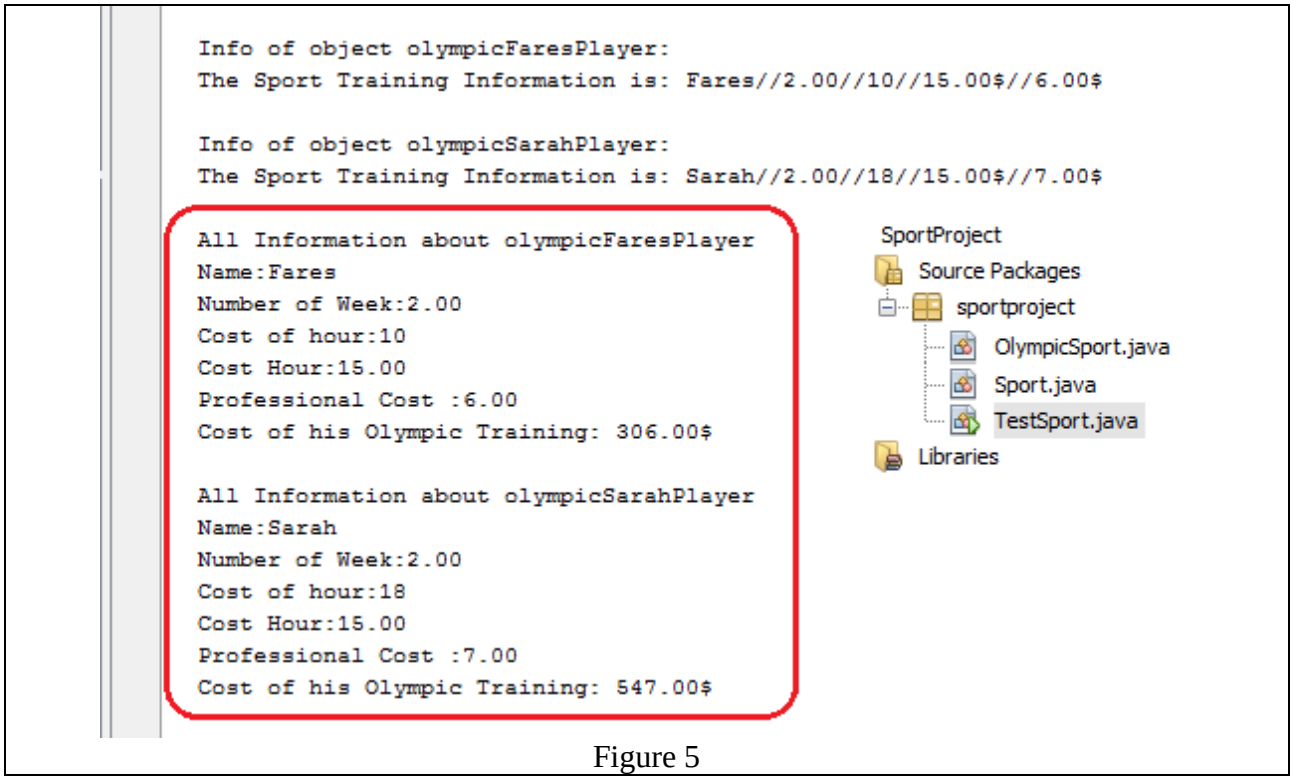

Figure 5