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

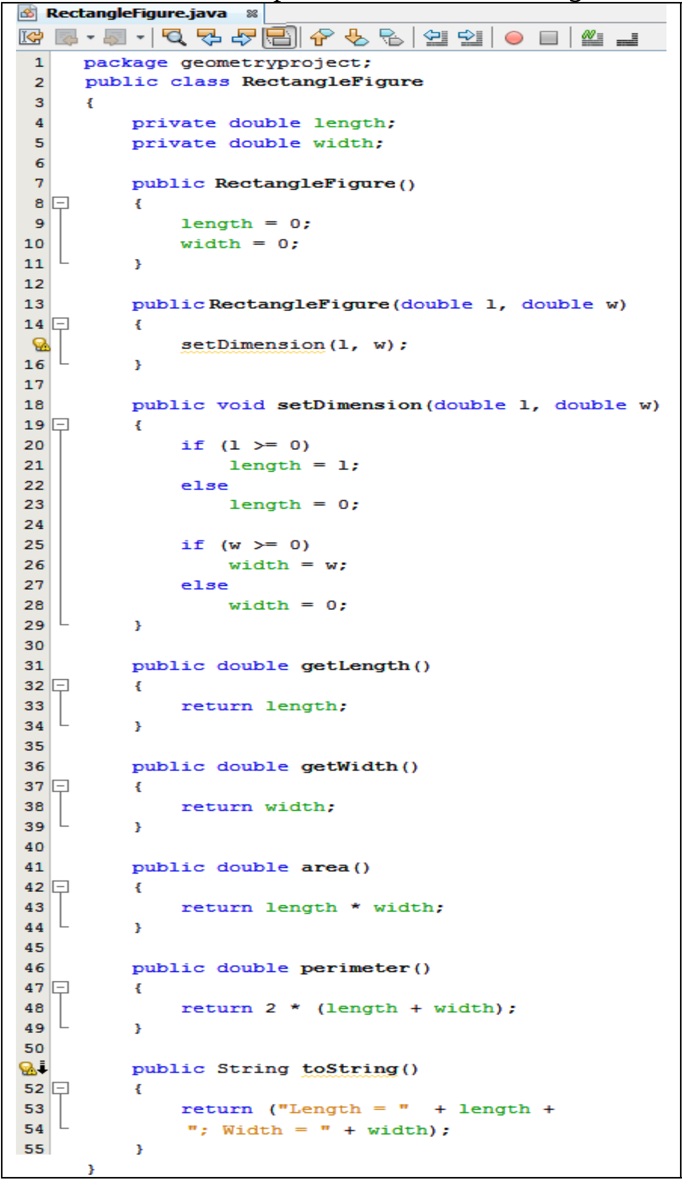
Teacher: Samir Chebbine **Lab 2** **Jul 04, 2022**

Lab 2: Polymorphism in Java

Complete all these following programs as explained in my **Lab 2 YouTube Video 1.** All *missing coding statements* are presented in this YouTube video with explanation.

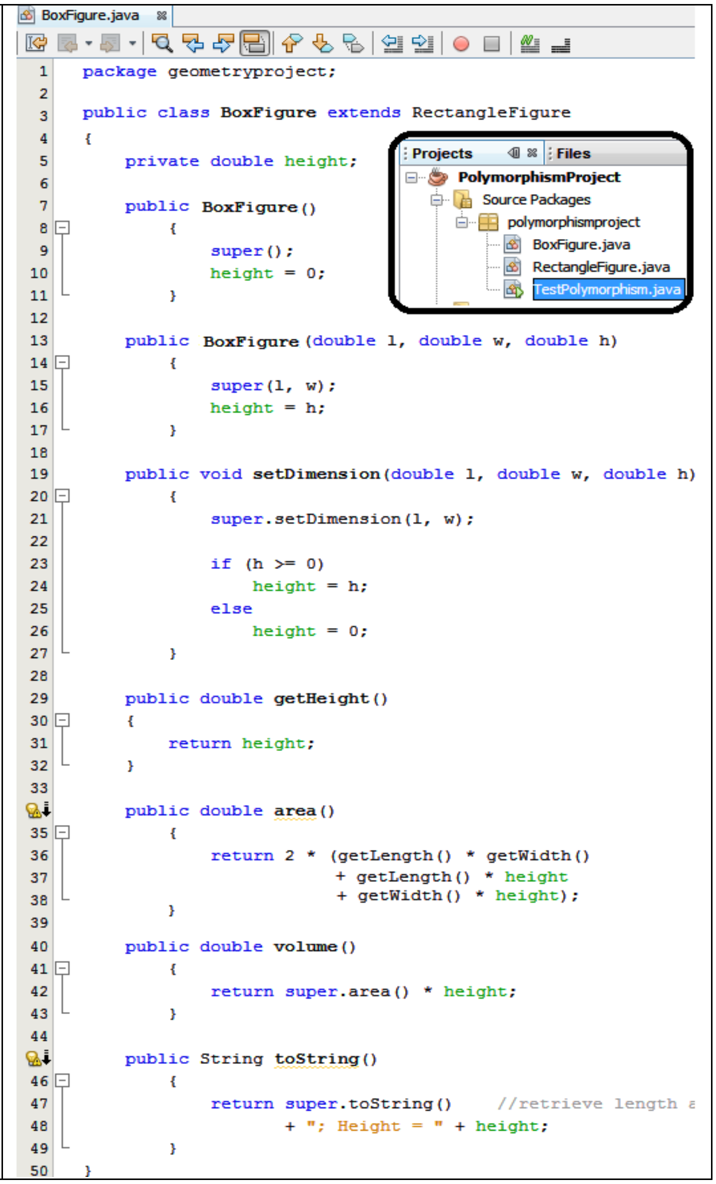
Create and Submit a Word file ***Lab2OOPProgramminIIYourName.doc*** which contains Answers of Book Exercises and output screenshots for every Java Project. Submit the Java projects too.

1. **Polymorphism**
   a) Create the Project **PolymorphismProject** shown hereafter. *RectangleFigure* class is called the *Superclass* and the *BoxFigure* class is called the *Subclass.*



RectangleFigure.java

```java
package geometryproject;
public class RectangleFigure
{
    private double length;
    private double width;

    public RectangleFigure()
    {
        length = 0;
        width = 0;
    }

    publicRectangleFigure(double l, double w)
    {
        setDimension(l, w);
    }

    public void setDimension(double l, double w)
    {
        if (l >= 0)
            length = l;
        else
            length = 0;

        if (w >= 0)
            width = w;
        else
            width = 0;
    }

    public double getLength()
    {
        return length;
    }

    public double getWidth()
    {
        return width;
    }

    public double area()
    {
        return length * width;
    }

    public double perimeter()
    {
        return 2 * (length + width);
    }

    public String toString()
    {
        return ("Length = "  + length +
        "; Width = " + width);
    }
}
```

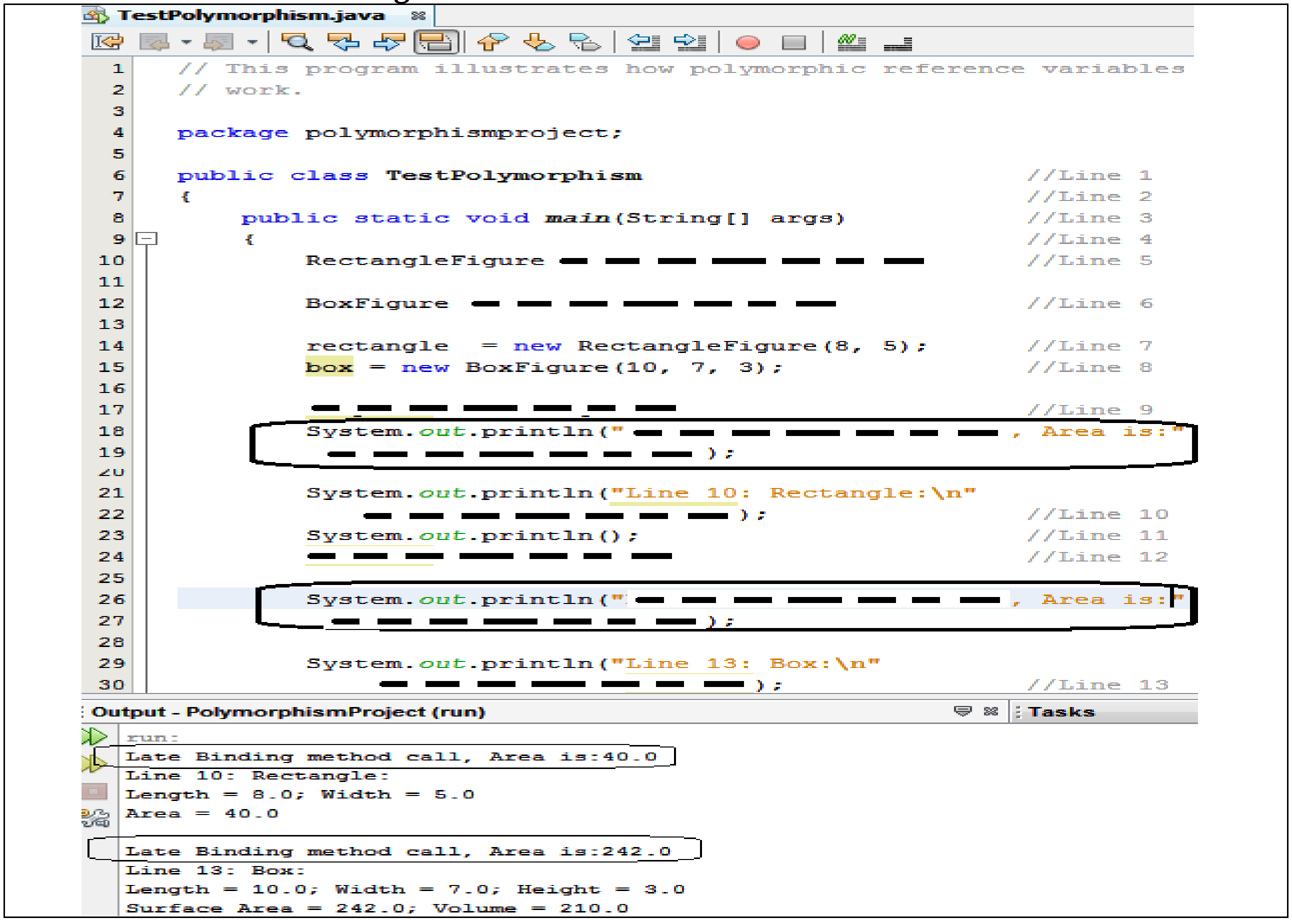


BoxFigure.java

```java
package geometryproject;

public class BoxFigure extends RectangleFigure
{
    private double height;

    public BoxFigure()
        {
            super();
            height = 0;
        }

    public BoxFigure(double l, double w, double h)
        {
            super(l, w);
            height = h;
        }

    public void setDimension(double l, double w, double h)
        {
            super.setDimension(l, w);

            if (h >= 0)
                height = h;
            else
                height = 0;
        }

    public double getHeight()
    {
        return height;
    }

    public double area()
        {
            return 2 * (getLength() * getWidth()
                  + getLength() * height
                  + getWidth() * height);
        }

    public double volume()
        {
            return super.area() * height;
        }

    public String toString()
        {
            return super.toString()    //retrieve length a
                + "; Height = " + height;
        }
}
```

(Testing the *RectangleFigure* and *BoxFigure* *classes*) Create the Java Program *TestPolymorphism.java* shown hereafter.

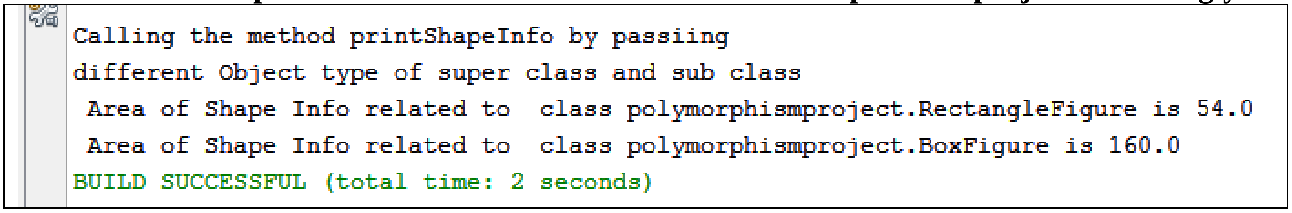
What is the Late Binding ?



```java
// This program illustrates how polymorphic reference variables
// work.

package polymorphismproject;

public class TestPolymorphism                                   //Line 1
{                                                               //Line 2
    public static void main(String[] args)                      //Line 3
    {                                                           //Line 4
        RectangleFigure __ __ __ __ __ __ __                    //Line 5

        BoxFigure __ __ __ __ __ __ __                          //Line 6

        rectangle  = new RectangleFigure(8, 5);                 //Line 7
        box = new BoxFigure(10, 7, 3);                          //Line 8

        __ __ __ __ __ __ __                                    //Line 9
        System.out.println(" __ __ __ __ __ __ __ , Area is:"
          __ __ __ __ __ __ __ );

        System.out.println("Line 10: Rectangle:\n"
           __ __ __ __ __ __ __ );                              //Line 10
        System.out.println();                                   //Line 11
        __ __ __ __ __ __ __                                    //Line 12

        System.out.println(" __ __ __ __ __ __ __ , Area is:"
          __ __ __ __ __ __ __ );

        System.out.println("Line 13: Box:\n"
           __ __ __ __ __ __ __ );                              //Line 13
```

```
run:
Late Binding method call, Area is:40.0
Line 10: Rectangle:
Length = 8.0; Width = 5.0
Area = 40.0

Late Binding method call, Area is:242.0
Line 13: Box:
Length = 10.0; Width = 7.0; Height = 3.0
Surface Area = 242.0; Volume = 210.0
```

b) **Add new method called printShapeInfo(RectangleFigure shapeObj) in Main Testing class as explained in Lab 2 YouTube Video 1 and complete the project accordingly.**

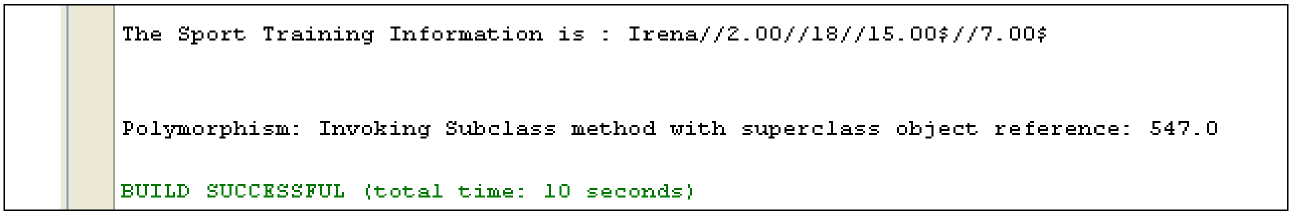


```
Calling the method printShapeInfo by passiing
different Object type of super class and sub class
 Area of Shape Info related to  class polymorphismproject.RectangleFigure is 54.0
 Area of Shape Info related to  class polymorphismproject.BoxFigure is 160.0
BUILD SUCCESSFUL (total time: 2 seconds)
```

2. Complete Project SportProject from Lab 1:

a) **(Polymorphism)**

- Within main class `TestSport`, call the method implementing the cost of training `CalculateCostTraining`*()* of the sub class **`OlympicSport`** with the super class object `yourPlayer` as shown hereafter.
- Test the polymorphism through other sub class objects.



```
The Sport Training Information is : Irena//2.00//18//15.00$//7.00$


Polymorphism: Invoking Subclass method with superclass object reference: 547.0

BUILD SUCCESSFUL (total time: 10 seconds)
```

3. **Answer the following questions:**

   a) Give an example of super class and sub class in **your own** stated project (different than Geometry, Sport projects).

   b) Define then data attributes of the specified super class. (give at least two data attributes)

   c) Define then data attributes of the specified sub class. (give at least one data attribute)

   d) Define then methods members of the specified super class. (at least one method)

   e) Define then an **overriding** methods member of the specified sub class. (at least one method)

   f) Write then a Java statement to instantiate an object from super class

   g) Write then a Java statement to instantiate an object from sub class

   h) Apply polymorphism with the super class object from `question (f)` to invoke the **overriding** method of sub class defined in `question (e)`. Write then the appropriate Java statements on how to use **polymorphism**.

4. **True or False and why:**

   a) It is Valid to write the following assignment in PolymorphismProject: `box = shapeRef;`

   b) It is Valid to write the following statements in PolymorphismProject:

```
RectangleFigure shapeRef;
shapeRef =rectangle2;//assume rectangle2 is an object of RectangleFigure class type
System.out.println("Late Binding method call, Volume is:" + shapeRef.volume());
```

   c) It is Valid to write the following statements in PolymorphismProject:

```
RectangleFigure shapeRef;

shapeRef =box2;  // assume box2 is an object of BoxFigure class type
System.out.println("Late Binding method call, Volume is:" + shapeRef.volume());
```

   d) Assume you have the method called `printShapeInfo(RectangleFigure shapeObj):` is the following call valid:

```
printShapeInfo(rectangle2); //rectangle2 is an object of RectangleFigure class type
```

   e) Assume you have the method called `printShapeInfo(BoxFigure shapeObj):` is the following call valid:

```
printShapeInfo(rectangle2); //rectangle2 is an object of RectangleFigure class type
```

   f) Assume you have the method called `printShapeInfo(RectangleFigure shapeObj):` is the following call valid:

```
printShapeInfo(box2); // box2 is an object of BoxFigure class type
```