King Saud University
College of Computer & Information Science
CSC111 – Lab07
Object – I –
All Sections
--

Instructions

Web-CAT submission URL:
http://10.131.240.28:8080/Web-CAT/WebObjects/Web-CAT.woa/wa/assignments/eclipse

Objectives:

- To describe objects and classes, and use classes to model objects.
- To use UML graphical notation to describe classes and objects.
- To demonstrate how to define classes and create objects.
- To create objects using constructors.
- To access objects via object reference variables.
- To define a reference variable using a reference type.
- To access an object's data and methods using the object member access operator (.).
- To define data fields of reference types and assign default values for an object's data fields.
- To distinguish between object reference variables and primitive data type variables.

Lab Exercise 1

Design a class named `MyRectangle` to represent a rectangle. The class contains:

- Two double data fields named `width` and `height` that specify the width and height of the rectangle. The default values are 1 for both width and height.
- A no-arg constructor that creates a default rectangle.
- A constructor that creates a rectangle with the specified width and height.
- A method named `getArea()` that returns the area of this rectangle.
- A method named `getPerimeter()` that returns the perimeter.

Draw the UML diagram for the class and then implement the class. Write a test program (class with main method) that creates two MyRectangle objects—one with width 4 and height 40 and the other with width 3.5 and height 35.9. Display the width, height, area, and perimeter of each rectangle in this order. Name rectangle class `MyRectangle`. Name class with main method `TestRectangle`.

Sample Run

```
The area of a rectangle with width 4.0 and height 40.0 is 160.0
The perimeter of a rectangle is 88.0
The area of a rectangle with width 3.5 and height 35.9 is
125.64999999999999
The perimeter of a rectangle is 78.8
```

Solution

1- First phase is to design your program as an OOP program. Draw UML diagrams for the two classes, MyRectangle and TestRectangle.

MyRectangle
width: double height: double

MyRectangle()
MyRectangle(newWidth: double, newHeight: double)
getArea(): double
getPerimeter(): double

TestRectangle
main(): void

2- Create a new eclipse project and name it `lab07`

3- Create a new class and name it `TestRectangle`. Make sure you choose the `public static void main` option. We will write both classes into this file, which means **MyRectangle** class, will not be public. Another option is to create two files one for each class.

4- Write the program as shown in next page (you can ignore comments)

5- When you are done, save your program and run it. Make sure it prints the output as shown above.
6- Submit your program to WebCAT through. Ask your TA for help.

```java
public class TestRectangle {
  public static void main(String[] args) {
    MyRectangle myRectangle = new MyRectangle(4, 40);
    System.out.println("The area of a rectangle with width " +
      myRectangle.width + " and height " +
      myRectangle.height + " is " +
      myRectangle.getArea());
    System.out.println("The perimeter of a rectangle is " +
      myRectangle.getPerimeter());

    MyRectangle yourRectangle = new MyRectangle(3.5, 35.9);
    System.out.println("The area of a rectangle with width " +
      yourRectangle.width + " and height " +
      yourRectangle.height + " is " +
      yourRectangle.getArea());
    System.out.println("The perimeter of a rectangle is " +
      yourRectangle.getPerimeter());
  }
}

class MyRectangle {
  // Data members
  double width = 1;
  double height = 1;

  // Constructor
  public MyRectangle() {
  }

  // Constructor
  public MyRectangle(double newWidth, double newHeight) {
    width = newWidth;
    height = newHeight;
  }

  public double getArea() {
    return width * height;
  }

  public double getPerimeter() {
    return 2 * (width + height);
  }
}
```

Lab Exercise 2

Design a class named `Stock` that contains:

- A string data field named `symbol` for the stock's symbol.
- A string data field named `name` for the stock's name.
- A double data field named `previousClosingPrice` that stores the stock price for the previous day.
- A double data field named `currentPrice` that stores the stock price for the current time.
- A constructor that creates a stock with the specified symbol and name.
- A method named `getChangePercent()` that returns the percentage changed from `previousClosingPrice` to `z`.

Draw the UML diagram for the class and then implement the class. Write a test program that creates a Stock object with the stock symbol, the company name, and the previous closing price read from user. Read a new current price from user and display the price-change percentage. Name your classes `Stock` and `TestStock`. Use two separate files for each of the two classes.

Sample Run

```
Enter symbol of stock:ORCL ↵
Enter company name:Oracle ↵
Enter previous closing price:34.5 ↵
Enter curret price:34.35 ↵
Previous Closing Price: 34.5
Current Price: 34.35
Price Change: -0.434782608695648%
```

Solution

1- First phase is to design your program as an OOP program. Draw UML diagrams for the two classes, Stock and TestStock.

Stock
symbol: String name: String previousClosingPrice: double currentPrice: double
Stock() Stock(newSymbol: String, newName: String) getChangePercent(): double getPreviousClosingPrice(): double getCurrentPrice(): double setCurrentPrice(newCurrentPrice: double): void setPreviousClosingPrice(newPreviousClosingPrice: double): void

TestStock
main(): void

2- Create a new eclipse project and name it `lab07`

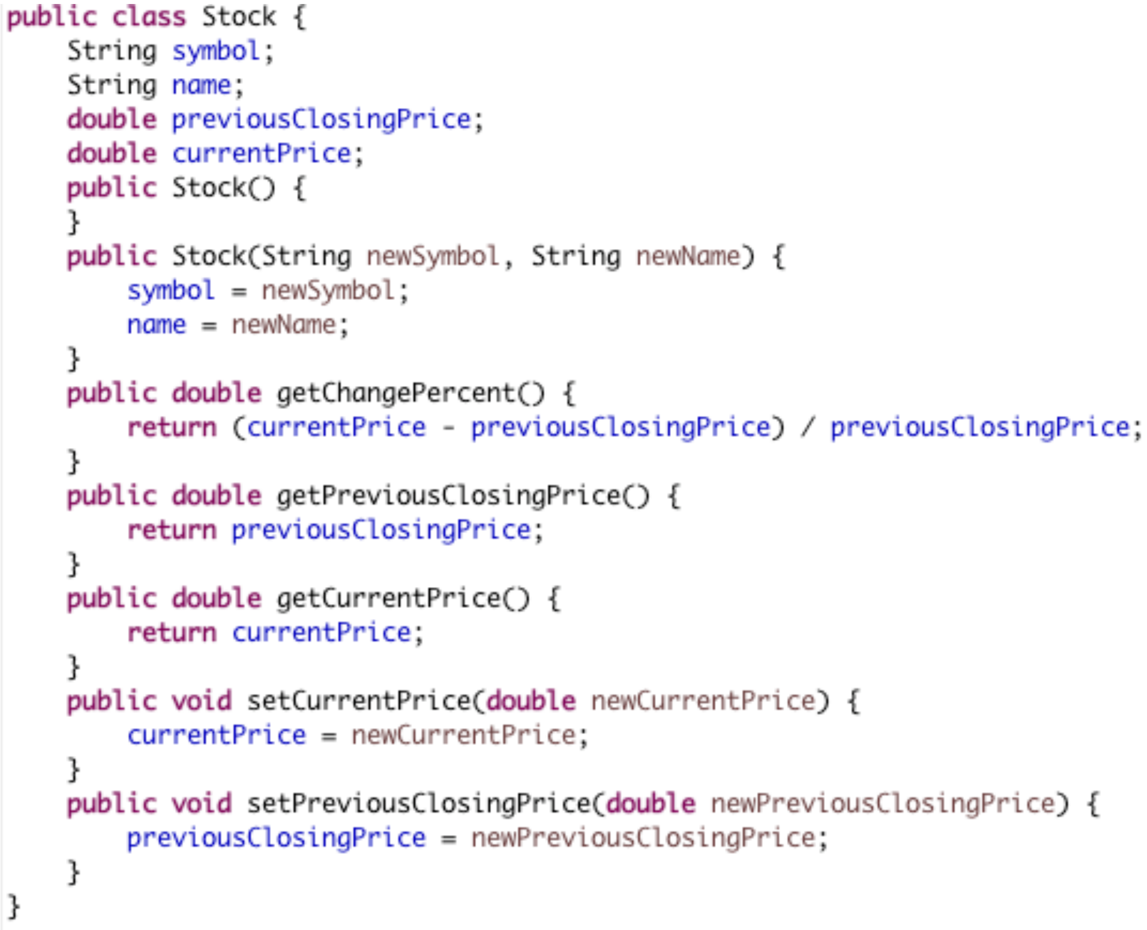
3- Unlike in previous exercise, we will create two separate files for the two classes. Create a new class and name it `Stock`.

4- Create a new class and name it `TestStock`. Make sure you choose the `public static void main` option.

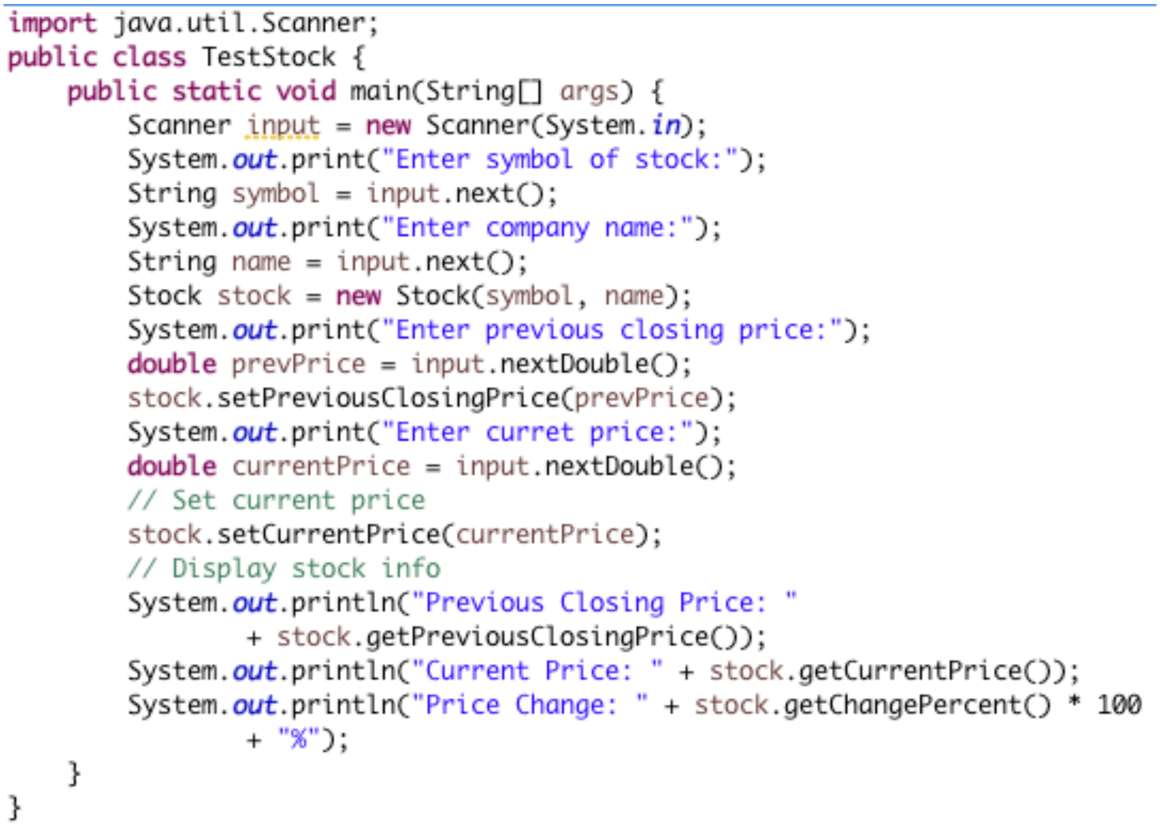
5- Write the two program classes as shown in next pages (you can ignore comments)

6- When you are done, save your program and run it. Make sure it prints the output as shown above.

7- Submit your program to WebCAT through. Ask your TA for help.



```
public class Stock {
    String symbol;
    String name;
    double previousClosingPrice;
    double currentPrice;
    public Stock() {
    }
    public Stock(String newSymbol, String newName) {
        symbol = newSymbol;
        name = newName;
    }
    public double getChangePercent() {
        return (currentPrice - previousClosingPrice) / previousClosingPrice;
    }
    public double getPreviousClosingPrice() {
        return previousClosingPrice;
    }
    public double getCurrentPrice() {
        return currentPrice;
    }
    public void setCurrentPrice(double newCurrentPrice) {
        currentPrice = newCurrentPrice;
    }
    public void setPreviousClosingPrice(double newPreviousClosingPrice) {
        previousClosingPrice = newPreviousClosingPrice;
    }
}
```



```
import java.util.Scanner;
public class TestStock {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter symbol of stock:");
        String symbol = input.next();
        System.out.print("Enter company name:");
        String name = input.next();
        Stock stock = new Stock(symbol, name);
        System.out.print("Enter previous closing price:");
        double prevPrice = input.nextDouble();
        stock.setPreviousClosingPrice(prevPrice);
        System.out.print("Enter curret price:");
        double currentPrice = input.nextDouble();
        // Set current price
        stock.setCurrentPrice(currentPrice);
        // Display stock info
        System.out.println("Previous Closing Price: "
                + stock.getPreviousClosingPrice());
        System.out.println("Current Price: " + stock.getCurrentPrice());
        System.out.println("Price Change: " + stock.getChangePercent() * 100
                + "%");
    }
}
```

Done…