**Lab 3: Object and Class II**

**Objectives:** To practice the knowledge of Object, Class

**Programming using Object-Oriented**

1. (The Stock class) Following the example of the Circle class, 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 **currentPrice**.
* Getter and setter for data fields **previousClosingPrice** and **currentPrice**

Draw the UML diagram for the class then implement the class. Write a test program that creates a Stock object with the stock symbol ORCL, the name Oracle Corporation, and the previous closing price of 34.5. Set a new current price to 34.35 and display the price-change percentage.

**Hint**: Use draw.io or other diagram drawing tools to create UML diagram. Show the UML diagram to TAs or lecturer to evaluate.

**Sample run**:

Text

Description automatically generated

--------------------------------------------------------------------Checkpoint 1-------------------------------------------------------------------

1. Write a class called **Counter** that represents a simple tally counter, which might be used to count people as they enter a room. The **Counter** class should contain a single integer as instance data, representing the count. Write a constructor to initialize the count to zero. Write a method called **click** that increments the count and another method called **getCount** that returns the current count. Include a method called **reset** that resets the counter to zero. Finally, create a driver class called **CounterTest** that creates two **Counter** objects and tests their methods. First **Counter** object call 2 **click** method, display number of its count, then call **reset** method and display number of count after reset. Second **Counter** object call 3 **click** methods, display number of its count, then call **reset** method and display number of count after reset.

Sample run:

Text

Description automatically generated

--------------------------------------------------------------------Checkpoint 2-------------------------------------------------------------------