North South University

Department of Electrical and Computer Engineering

CSE 215L: Programming Language II Lab

Lab Manual - 7

Lab Instructor: Taif Al Musabe

Objective:

- To describe objects and classes, and use classes to model objects
- To access an object's data and methods using the object member access operator (**.**)
- To define private data fields with appropriate getter and setter methods
- To encapsulate data fields to make classes easy to maintain
- To use the keyword **this** to refer to the calling object itself

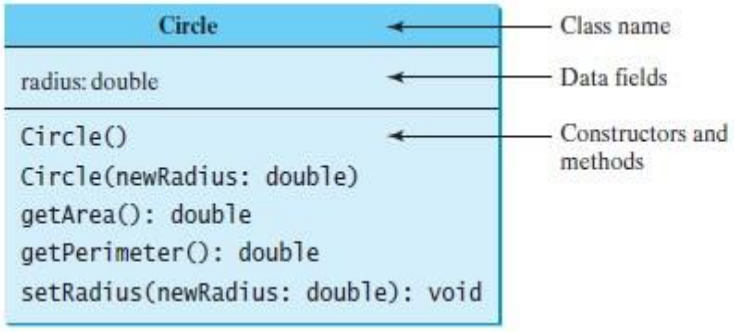
```
class Circle {
/** The radius of this circle */
  double radius = 1;
/** Construct a circle object */
  Circle() {
  }
 /** Construct a circle object */
  Circle(double newRadius) {
    radius = newRadius;
  }
/** Return the area of this circle */
double getArea() {
    return radius * radius * Math.PI;
  }
/** Return the perimeter of this circle */
double getPerimeter() {
      return 2 * radius * Math.PI;
  }
/** Set new radius for this circle */
double setRadius(double newRadius) {
      radius = newRadius;
   }
 }
```

Data field

Constructors

Method

UML Class Diagram



Task – 1

(*The **Rectangle** class*) 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.

Write a test program that creates two **Rectangle** objects and display the width, height, area, and perimeter of each rectangle.

Task – 2

(*The **Triangle** class*) The class contains:

■ Two **double** data fields named **base** and **height** that specify the base and height of the rectangle. The default values are **1** for both **base** and **height**.

■ A no-arg constructor that creates a default triangle.

■ A constructor that creates a rectangle with the specified **base** and **height**.

■ A method named **getArea()** that returns the area of this rectangle.

■ A method named **showWidthOfEachSide()** that print each side of the triagnle.

Write a test program that creates two **Trinagle** objects. Display all sides, and area of each triangle.

Homework – 1

(*The **Stock** class*) Following the example of the **Circle** class in Section 9.2, 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 **currentPrice**.

Write a test program that creates a **Stock** object with the stock symbol, the name and the previous closing price. Set a new current price and display the price-change percentage.

Homework – 2

(*The **Account** class*) Design a class named **Account** that contains:

■ A private **int** data field named **id** for the account (default **0**).

■ A private **double** data field named **balance** for the account (default **0**).

■ A private **double** data field named **annualInterestRate** that stores the current interest rate (default **0**). Assume all accounts have the same interest rate.

■ A private **Date** data field named **dateCreated** that stores the date when the account was created. ■ A no-arg constructor that creates a default account.

■ A constructor that creates an account with the specified id and initial balance.

■ The accessor and mutator methods for **id**, **balance**, and **annualInterestRate**.

■ The accessor method for **dateCreated**.

■ A method named **getMonthlyInterestRate()** that returns the monthly interest rate.
■ A method named **getMonthlyInterest()** that returns the monthly interest.
■ A method named **withdraw** that withdraws a specified amount from the account.
■ A method named **deposit** that deposits a specified amount to the account.

(*Hint*: The method **getMonthlyInterest**() is to return monthly interest, not the interest rate. Monthly interest is **balance** * **monthlyInterestRate**. **monthlyInterestRate** is **annualInterestRate** / **12**. Note that **annualInterestRate** is a percentage, e.g., like 4.5%. You need to divide it by 100.)

Write a test program that creates an **Account** object with an account ID, a balance , and an annual interest rate. Use the **withdraw** method to withdraw some money, use the **deposit** method to deposit some, and print the balance, the monthly interest, and the date when this account was created.