**A close up of a sign

Description automatically generated**

*Fall 2021*

**CSC/BIF 243**

**Introduction to Object Oriented Programming**

*Course Instructor: Joe Khalife*

Lab11

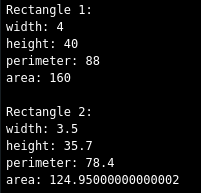
**Problem 1:**

Implement a class named **Rectangle** to represent a rectangle. The class contains:

* Two data fields named **width** and **height**.
* A constructor that creates a rectangle with the specified **width** and **height**. The default values are **1** and **2** for the **width** and **height**, respectively.
* 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—one with width **4** and height **40** and the other with width **3.5** and height **35.7**. (you can read them from the user or just put them directly in your code) Display the width, height, area, and perimeter of each rectangle in this order (in the main. Just create 2 rectangles instances and call the 4 functions on each of the rectangles. And print their results)

**Output will look like this :**



**Problem 2**

Implement a class named **Stock** to represent a company’s stock that contains:

* A private string data field named **symbol** for the stock’s symbol.
* A private string data field named **name** for the stock’s name.
* A private float data field named **previousClosingPrice** that stores the stock price for the previous day.
* A private float data field named **currentPrice** that stores the stock price for the current time.
* A constructor that creates a stock with the specified symbol, name, previous price, and current price.
* A get method for returning the stock name.
* A get method for returning the stock symbol.
* Get and set methods for getting/setting the stock’s previous price.
* Get and set methods for getting/setting the stock’s current price.
* 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 INTC, the name Intel Corporation, the previous closing price of **20.5**, and the new current price of **20.35**, and display all the information of the newly created Stock along with the price-change percentage.

**Output:**

**Intel cooperation**

**INTC**

**20.5**

**20.35**

**-0.7317073170731638**

**Problem 3:**

Implement a class named **Account** that contains:

* A private **int** data field named **id** for the account.
* A private float data field named **balance** for the account.
* A private float data field named **annualInterestRate** that stores the current interest rate.
* A constructor that creates an account with the specified id (default 0), initial balance (default 100), and annual interest rate (default 0).
* The accessor and mutator (getters/setters) methods for **id**, **balance**, and **annualInterestRate**.
* 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 the monthly interest amount, not the interest rate. Use this formula to calculate the monthly interest: **balance \*monthlyInterestRate**. **monthlyInterestRate** is **annualInterestRate/ 12**.

Note that **annualInterestRate** is a percent (like 4.5%). You need to divide it by **100**.)

Write a test program that creates an **Account** object with an account id of 1122, a balance of $20,000, and an annual interest rate of 4.5%. Print the information of the newly created object.

Now, use the **withdraw** method to withdraw $2,500, use the **deposit** method to deposit $3,000, and then print the id, balance, monthly interest rate, and monthly interest after doing the withdraw and deposit.

**Output:**

**4.5**

**20000**

**1122**

**0.375**

**75.0**

**After doing the withdraw and deposit .**

**4.5**

**20500**

**1122**

**0.375**

**76.875**