USMAN INSTITUTE OF TECHNOLOGY

Department of Computer Science
CS121 Object Oriented Programming

Lab # 07
Data Hiding

Objective:

This experiment introduces the students to the concept of Data Hiding using Python as a Programming Language. The concept of public and private data members and role of accessors and mutators is explained

Name of Student: _____

Roll No: _____ **Sec.** _____

Date of Experiment: _____

Marks Obtained/Remarks: _____

Signature: _____

Student Exercise

The **Fan** class

Design a class named Fan to represent a fan. The class contains:

■ Three constants named SLOW, MEDIUM, and FAST with the values 1, 2, and 3 to denote the fan speed.

■ A private int data field named speed that specifies the speed of the fan.

■ A private bool data field named on that specifies whether the fan is on (the default is False).

■ A private float data field named radius that specifies the radius of the fan.

■ A private string data field named color that specifies the color of the fan.

■ The accessor and mutator methods for all four data fields.

■ A constructor that creates a fan with the specified speed (default SLOW), radius (default 5), color (default blue), and on (default False).

Exercise 1

Draw the UML diagram for the class and then implement the class.

Exercise 2

Write a test program that creates two Fan objects. For the first object, assign the maximum speed, radius 10, color yellow, and turn it on. Assign medium speed, radius 5, color blue, and turn it off for the second object. Display each object's speed, radius, color, and on properties.

The **Account** class

Design 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 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.

Exercise 3

Draw the UML diagram for the class, and then implement the class.

(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.)

Exercise 4

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%. Use the withdraw method to withdraw $2,500, use the deposit method to deposit $3,000, and print the id, balance, monthly interest rate, and monthly interest.