King Hussein School for Computing Sciences
Department of Computing Science

Object Oriented Programming Lab
Assignment #3 – Classes and data abstraction (1)

Instructor: Bushra Alhijawi   Fall 2020/2021

Student Name	
Student Univ. ID	
Date	

	Q1 (60)	**Q2 (40)**
100%		
90%		
75%		
50%		
0%		

Total (100):________

Grading Criteria

- **100%**: Program is fully correct, code is well-indented, identifiers are well-named and the output is well formatted.
- **90%**: Program is fully correct but meaningless identifier names are used, indentation is bad, or output is not well formatted.
- **75%**: Program runs mainly correctly. It fails to adhere to some of the specifications.
- **50%**: Program contains syntax errors but seems correct.
- **0%**: Program has serious syntax and/or logical errors.

Lab Objectives

- Learn how to define a simple class and implement its member functions.
- Learn how to use getters and setters
- Learn how to use a user-defined class in an application.

Lab Instructions

- Create a new folder on desktop and name it by your "ID-FirstName-LastName-Lab1". Make sure to save your solutions in this folder.
- You must upload your solutions to e-learning as follows:
  - **Each project** you create it as a .zip file.
  - A **text file** associated with each project contains a copy of the code.

Lab Exercises

This assignment consists of two exercises.

Exercise 1 – Account Class

Exercise Objectives

- ✓ Define class
- ✓ Declare objects

Problem Description

Inside "***Lab3***"folder, create a project named "***Lab3Ex1***". Use this project to write and run a C++ program that consists of class called **Account**:

- ➢ This class includes the following private members:
    - int id
    - string name
    - double balance
- ➢ This class includes the following public members:
    - Setter functions to set the values of the three private members.
    - A function called **getId** that returns the id.
    - A function called **getname** that returns the name.
    - A function called **getbalance** that returns the balance.
    - A function called **credit** that receives a double value (amount) and add the amount to balance, return balance.
    - A function called **debit** that receives a double value (amount) and subtract the amount from balance if the amount <=balance. Otherwise, print "Amount Exceeded". This function returns balance.
    - A function called **transferTo** that transfer the amount from the account to the other account if the amount <=balance. Otherwise, print "Amount Exceeded".

      transferTo(Account account, double amount)

Exercise 2 – Driver Program

Exercise Objectives

- ✓ Define class
- ✓ Declare objects

Problem Description

Write the **main** function that uses **Account** class as follows:

- Declare two objects, **account1** and **account2**. Initialize **account1** with id=1234 and name="Ahmad". Initialize **account2** with id, name, and balance that are read by a user.

  Hint: you can use #include<string> and getline(cin, str) to deal with string variables.
- Set the balance of **account1** to 1520.
- Print the balance of the two accounts using getters functions. You are supposed to print it as follows:
- Use **credit** function to add an amount entered by the user to the **account1**.
- Use **debit** function to withdraw 750$ from **account2**.
- Use **debit** function to withdraw 1750$ from **account1**.
- Print the balance of the two accounts using getters functions.
- Use **transferTo** function to transfer a specified amount by the user from **account1** to **account2**. Print the balance of the two accounts using getters functions.

```
Please enter the id, name, and balance:
2535
Ali
3250
************************************
Customer 1234, Ahmad has 1520$
Customer 2535, Ali has 3250$.
************************************
Please enter the credit amount
200
Your balance is 1720$
************************************
Amount Exceeded!!!
Customer 1234, Ahmad has 1720$
Customer 2535, Ali has 2500$.
************************************
Please specify the amount that you want to transfer
500
************************************
Customer 1234, Ahmad has 1220$
Customer 2535, Ali has 3000$.
```