**OOP Lab07 – Bank Account**

The goal of this lab isto create a class that models a bank account. The BankAccount class is fairly simple – it has two properties: a String variable that stores the name of the account owner and a double variable that stores the current balance of the account. These two fields (instance variables) are what **describe**a BankAccount*.*

The BankAccount class should have the following:

1. Two private instance variables (fields), String name and double balance.
2. A constructor that has two parameters: a String n and a double bal. The constructor should initialize the fields to the values of the parameters.
3. public void deposit(double amt) – this method doesn’t return anything, and has one double variable parameter. The parameter represents the amount of money that will be added to the account’s balancefield.
4. public void withdraw(double amt) – this method doesn’t return anything, and has a double variable parameter. The parameter represents the amount of money that will be withdrawn from the account’s balancefield.
5. Include an overridden toString() method that returns a String representation of the account object in the format: “Fred ($300.23)” (The balance should be displayed with typical currency precision.)
6. Next,create a new class, BankAccountRunner, with a main() method. Inside the main method, you should create 3 new BankAccount objects (remember, objects are a *specific instance* of a class). Initialize the BankAccount objects with whatever names and values you want.

Creating a new object follows this format:

ClassName objectName = new ClassName(<constructor's parameters>)

Using the BankAccount objects you created, write a bank teller user interface (called BankAccountRunner) that functions as shown in this video: <https://www.youtube.com/watch?v=_T3uDWHjnAQ>

**User interface appearance should be EXACTLY like in the video. Don’t cut corners.**

**To make the TestGUI happy, make sure you start off with these 3 accounts:**

* **McCoy $1000000**
* **Jim $600.45**
* **Bob $0.32**