**Skill Check Lab D Requirements**

**Overall Theme**: Tracks bank account information using the Account class as the parent class to the Checking and CD classes.

**Project**: The project is contained in the folder SkillCheckD.

**Overall Goal of this Skill Check:**

* Implement a singleton class that can only have one instance, that can be used throughout the application for the purpose of synchronizing.
* Create and start separate threads to apply the interest in each account.
* Catch any exceptions that are raised in the method called by the secondary thread.
* Synchronize the code in the method called by the secondary thread.
* Make sure all secondary threads are completely done before the event handler ends.

**Project Requirements**:

**Refer to the TODO comments in the project for specifics.**

**NOTE:**

* This project reads two separate input files to provide data for your project. The code is already implemented to accomplish this task.
* Then interest rates are applied when the Bank class raises an event to the BankAccounts (contains the collection of all accounts) class. The interest rates are stored in the decimal array in the Bank class and already loaded. There will be two times the interest rates will be applied. The first array element is applied the first time to all the accounts and the second array element is applied the second time to all the accounts. There is a sleep method to pause 2 seconds in between each interest rate application to similate the real world of a monthly application of interest to accounts. The event is raised each time by the Bank class.
* The event handler in the BankAccounts class calls the CalculateInterest method in each account passing in the interest rate to be applied. This call will be made on secondary threads.
* The code in the CalculateInterest method for each account type needs to be synchronized. The balance in each account is a common resource in each account that can be accessed by different threads causing a racing condition. The Console window is also a common resource.
* Also, all accounts need to be updated before their information is printed. So, all secondary threads must be completely done before the main thread proceeds to print out the account information.

**When completed, do as follows**:

1. Run the application one more time and notify the instructor when you are done, leaving the Console window open. DO NOT PRESS the ENTER key until after the instructor sees the output on the Console window.