C# Objects

Objects

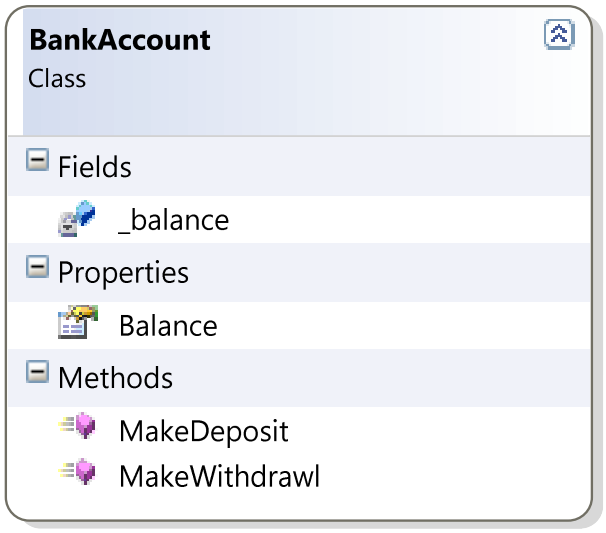
Exercise 1

Using object-oriented design

Scenario

Create a banking application.

Tasks	Detailed Steps
1. Setting up the lab	**a.** Start Visual Studio and create a new console application project name nyu.lab3.1 **Note:** For convenience, you should save the project to your flash drive.
2. Requirements	**a.** You will build a rudimentary banking application as a console application **b.** Create helper methods so that your code is modular. **c.** Be sure you include error handling to protect against illegal user input. **d.** The application should allow a user to • Create a customer bank account • Make a deposit • Make a withdrawl • Make a balance inquiry

3. Create the class	**a.** Create a class called BankAccount. You will write your source code in a new class file of the same name. It is a good practice to define every class in its own source file, in this case BankAccount.cs will hold the definition of the BankAccount class. **b.** The BankAccount class has the following members:  **Note:** Remember to make your fields private, all other members, including the class, should be given an access modifier of "public"
4. Business Rules	**a.** The BankAccount class should protect access to the _balance field. It can only be modified by the MakeDeposit or MakeWithdrawl methods. The Balance property should be read-only, you should not be able to set the balance to a new value. **b.** The MakeDeposit method should ignore amounts <= 0. **c.** The MakeWithdrawl method should throw an ApplicationException if the amount is > the balance.
5. Crate the program	**a.** In the Progam.cs file's Main method, create a collection of 3 BankAccount instances. **b.** Populate the objects with values. **c.** Prit out the object values.

6. Taking it further	**a.** Create a constructor method which accepts an initial balance. **b.** Add an account number property. Modify the constructor so that a BankAccount object can only be created (instantiated) when an account number is provided **c.** Create a SavingsAccount class which extends BankAccount. It will keep up with a savings goal and provides methods which report the savings goal, modify the goal, and tell you how the current balance compares against the goal.