

## Project 4 – Classes and Objects

The primary purpose of Project 4 is to help interpret the meaning and the use of Objects/classes within objects. An object in Java (and in Object-Oriented Programming) is a data structure or function that contains specific data to itself. An object is referenced by an identifier, such as a variable instantiated to be an object:

```
Object objectName = new Object();
```

A class is the body of an object that holds the methods of the object. Classes may contain everything from global variables referenced by the object to methods and functions that make the object unique.

In this project, you will be building a bank account that holds the user's basic information. To start, create a class `Project4` in your **default** package. This will be used to house the main function. Next, create a package named `APTest`. Within this package, create the class `BankAccount`. This will be your bank account object.

Start with your `BankAccount` class. The class should contain the following instances:

- Private string instance variables named `name`, `accoType` (account type – checking, savings)
- Private integer variable named `id`
- Private double variables named `balance` and `interest`

Next, create the following methods:

- Constructor with **NO** parameters that sets the `name` and `accoType` to `null`, and the `ID`, `balance` and `interest` to `0`
- Constructor with parameters for all the variables (i.e. – `name`, `accoType`, etc) and set the variables equal to their corresponding parameter
- Getters and setters for all 5 variables (each variable should have one getter function and one setter function, meaning you will be making 10 methods total here)
- `public void deposit (double amount)` – will take `amount` as a parameter and add that amount to the `balance`
- `public void withdraw (double amount)` – will take `amount` as a parameter and subtract that from the `balance`. Should also use the `zeroBalance()` function (see below) to make sure the account has money in it. It should also check to see if the amount is greater than the `balance` so the user doesn't overdraw their account.
- `public void monthlyPayment()` – subtracts the monthly interest payment, which is:
  - `balance = balance - balance * interest` OR `balance = balance * (1 - interest)`
- `public void edit()` – Prompts the user with input/output (will need a `Scanner`) to set the variables in your class.
- `Public String toString()`

- `private boolean zeroBalance()` – Checks to see if the balance is less than or equal to 0 and returns true if it is

The main function should be an implementation of this class. You should create a `BankAccount` object and use the above functions to show how it works.

Side Objectives:

1. You should try implementing a menu. This will help with your switch statements. Complete the above project and then work on the menu part of it.
2. You should use formats to format your string output. For example, you'll want your percent to print out as a percent, and your balance to print out as a currency. The way you do this is through `java.util.NumberFormat`. Complete the project and I will send you information on how to accomplish this. Good luck!