# COMP 1516 Assignement#3

* In this assignment, you will create a class called BankAccount
* Ensure that every method is named as a verb, has Doc- String, and follow all the style guidelines
* Create two Python files named BankAccount.py and main.py
* Add a comment that contains your name and student number at the top of each file

The functionality of each file is described below:

## Class BankAccount

This class has two instance variables:

* name
* balance
* Create a constructor that takes two parameters to initialize the instance variables name and balance. The name must not be an empty string. The name also must not include any numbers or symbols. The balance must be zero or more to be used.
  + If the name was an empty string a Value Error will be raised with the message “name cannot be empty”
  + If the name contained numbers or symbols a Value Error will be raised with the message “name cannot include numbers or symbols”
  + If the name was valid, it will be formatted properly and assigned to the instance variable.
    - The valid name consists of alphabetical characters lowercase or upper case, may include space, underscore or hyphen.
    - The name may consist of first, middle and last name.
    - One way to validate the name is using Regular Expressions.
    - Proper name format is the first character uppercase and the rest of the name lowercase character. For example, if the value “joHn jAcK bLaCk” was given the name instance variable would hold “John Jack Black”.
  + If the balance was less than zero a Value Error will be raised with the message “balance cannot be negative”
  + If the balance was 0 or a positive number it will be assigned to the instance variable balance
* Create method displayDetails(), the method will display the name and the balance in an organized manner
* Create method deposit(), the method accepts amount as a parameter. If the passed parameter was positive it will be added to the balance otherwise, a Value Error will be raised with the message “amount cannot be zero or less”
* Create method withdraw(), the method accepts amount as a parameter.
  + If the passed parameter was negative or zero a Value Error will be raised (“ amount cannot be zero or less than zero”
  + If the passed parameter was greater than balance a Value Error will be raised “Insufficient funds”
  + If the amount was valid i.e. greater than 0 and less or equal to the balance, the amount will be deducted from the balance

## main.py

This file will import BankAccount.py and create the main method

The main method will handle all the errors in a try-except block

The main method will implement the following

* Ask the user to input name and balance and use them to create a BankAccount object.
  + If some or all are values were invalid the user will be asked again to input the name and the balance
  + If all the provided values were valid, the loop will be terminated and the values will be used to create a BankObject.
* The main method will display the following choices:
  1. deposit
  2. withdraw
* The user will be asked to choose the transaction type and the transaction will be completed.
  + If an Error was raised, the user will be notified and he/she will be asked if he/she wants to continue
  + If the transaction was successful the details of the BankAccount object will be displayed
* The user will be asked if he/she wants to make another transaction
* The process will be repeated until the user chooses to terminate the loop

## Sample output

The assignment output should be similar to the output provided in file sample\_output.txt

# Submission

You will submit a zipped folder that has the following files:

* BankAccount.py
* main.py

Your instructor may use different data to check your work. Create your code accordingly.

Marks will be given for functionality and style.

Assignments that do not compile or throw an error will receive 0 marks

Upload your zipped folder to the dropbox before the deadline posted on the dropbox