**ITEC 136 Lab 4 Name: \_\_Danielle Hooven\_\_\_\_\_\_**

## Assignment

Please update the following Python program to include additional methods:

* 1. deposit(500) #Adds $500 to the account
  2. withdraw(500) #Takes $500 from the account
  3. transfer (b, 500) #Transfers $500 from the account to account b. Account b is bank account object.
  4. transactions() #Prints all the transactions line by line.
  5. overdraft\_fees() #Prints the sum of overdraft fees. Assume $35 per one.
  6. another\_method() #Create another method that you think necessary and change the name appropriately

Test each method comprehensively before submission. For example, you cannot transfer $300 to another account, if your account has only $100. **If there is insufficient funds, do not update the balance. Treat the overdraft fee as a separate fee.** **Do not deduct the overdraft fee from the balance.**

**Hint:** You will need to add more object properties.

|  |
| --- |
| **class** BankAccount():  **def** \_\_init\_\_(self, name, account\_number, initial\_amount):  self.name = name  self.no = account\_number  self.balance = initial\_amount  **def** deposit(self, amount):  self.balance += amount    **def** status(self):  s = '{}, {}, balance: {:.2f}'.format(self.name, self.no, self.balance)  **print(**s) |

## Grading Criteria

* Quality of the solution: 0 – 50 points
  + Documentation (internal comments) (0 – 5 points)
    - It is not uncommon for industry to impose a way of documenting software that is uniform across all programmers. Documentation standards are outlined in the same document referenced in the style bullet below.
  + Style (formatting, naming conventions and appearance of solution code) (0 – 5 points)
    - As listed in the action items of the course web page lab assignment, ensure that your solution code meets the documentation and style guidelines for this course. The documentation and style requirements Word document titled "DocumentationAndStyleGuidelines.doc” exists as a link on the lab assignment page. Review your code against this document.
  + Correctness (how the program works under test) (0 – 40 points)
    - The program should work, and be seen to work. The code should be robust.
* Note that grammar, spelling, and writing mechanics do not contribute to the total points for this assignment; however, improper grammar, poor spelling, or poor writing mechanics may result in significant point deductions.