CSCI 4041, Fall 2018, Programming Assignment 1
Due Friday, 2/1/19, 1:00 PM (submission link on Canvas)

This is not a collaborative assignment; you must design, implement and test the solution(s) on your own. You may not consult or discuss the solution with anyone other than the course instructor or TAs. In addition, you may not include solutions or portions of solutions obtained from any source other than those provided in class. Obtaining or sharing solutions to any programming assignment for this class is considered academic misconduct. If you are not sure what this means, consult the class syllabus or discuss it with the course instructor.

Introduction:

You are r0b1n h00d, super hacker. Your objective is to take from the rich and give to the poor. Specifically, you have found a way to access all of the accounts at various banks. Your plan is simple: at each bank, move $1337 from the account with the maximum balance to the account with the minimum balance. You don't see any way this could possibly go wrong.

The purpose of this assignment is to check your understanding of basic Python programming, especially control structures, objects, functions, and lists. Mastery of Python is not necessary to succeed in CSCI 4041, since programming is not the focus of the class; if you understand how to use the concepts listed above, you should be fine. This should be a very straightforward assignment, so If you find yourself struggling, I encourage you to ask for Python help during office hours or via email.

To complete this assignment, you'll need to download Python 3. You can do this here: https://www.python.org/downloads/. The version shouldn't matter so long as it's Python 3.X. There are many tutorials out there to help you learn the basics of Python if you have somehow avoided it in your previous coursework: feel free to look around and find one that best suits you. Python even has an official tutorial included in its documentation: https://docs.python.org/3/tutorial/. Again, a lot of this won't be relevant to the class: we're pretty much only concerned with control structures, lists, functions, and objects.

Instructions:

Download the PA1.py template from the course website.

The template contains an `Account` class, which represents a single bank account. Bank accounts have two instance variables:

- `.name` is a string representing the name of the account holder.
- `.balance` a floating point number representing the balance of the account (which may be positive or negative), in dollars.

You must implement the `hack_bank` function. This function takes one argument: `account_list`, which is a list of `Account` objects, representing all of the accounts for a given bank. If there are at least two `Account` objects in account_list, this function should find the account with the minimum balance (that is, the most negative balance, or least positive if there are no negative balances), add $1337 to it, and then subtract $1337 from the account with the maximum balance.

You can assume that there will be no ties for the minimum/maximum balance. If there are less than two `Account` objects in account_list, the function should do nothing. The function should not return anything, and should not alter the names of the accounts in the list, or the order in which they appear.

Requirements:

- You must download the template file PA1.py and edit the `hack_bank` function. You can create your own helper functions, but don't edit the code beyond the "DO NOT EDIT" line.
- You are not permitted to use the `input()` function as this will break the grading script, nor can you import any modules beyond the ones already imported in the template.
- You may use any strategy you want to complete the task, so long as it works for any reasonable list of `Account` objects and doesn't break the above requirements.
- Your program must run without errors on the version of Python installed on the CSELabs machines, Python 3.5.2. (if you're testing this on CSELabs, you need to type python3 or idle3 instead of python or idle to start the correct version from the terminal)
- This assignment will be graded automatically based on the number of test cases your program passes. There will be several secret test cases in addition to the ones included in the template to ensure you're not hard-coding in the solutions.
- This program will only run test cases until you fail one, avoiding the problem of having to scroll through test output to find the one broken test case.
- The grading breakdown for this assignment is as follows:
  - 30%: File runs without syntax errors
  - 70%: Passing test cases without breaking any requirements.
- The unedited template file already runs without syntax errors and passes 2 test cases. This means that if your program causes syntax errors or passes less than 2 test cases, you will get a better score by just submitting the original template unedited.
- Submit your edited PA1.py file to the Programming Assignment 1 link on Canvas before 1:00 PM on 2/1/19. No credit will be given for late submissions.