CMPS 4143

Programming Assignment-4 (Due November 10th 11:59 PM)

[*To get the full credit of your programming assignment, find the problem solving steps and documentation guidelines listed in 'Programming Assignments' section in D2L. You are supposed to submit a zip file (don't use winrar to zip) that should contain a documentation file like a docx file and your codes with proper programming extensions like .java or .py. Make sure all of your code is compliable or no runtime error. TA will not grade if your code is not runnable and you will get a zero; Sometimes your grade determines based on how good you can describe/explain the code to TA and Instructor***]**

1. (35 points) Write a Python program using file operation. You will open an input file "students.dat" that will contain a list of student names, classification, and grade in the class. (All student info is completely made up) You should read through the entire input file. After reading in all information, do operations (**No built-in functions like Average, min, max, count, etc.**), close the input file and write that following information with labels to an output file "student_statistics.txt"
   - Highest grade in the class
   - Lowest grade in the class
   - Class average grade (rounded to one decimal place)
   - Number of freshmen students
   - Number of sophomore students
   - Number of junior students
   - Number of senior students

   Sample contents inside the **input** file "Students.dat"; each of the following refers to a single line, i.e.; 6 lines in our input file:

   Holly Berry freshman 88

   Red Johnson sophomore 74

   Jeff Bozo freshman 91

   Pebble Johnston senior 82

   Thomas BradyBunch freshman 63

   Eddy Sheen junior 97

Sample content inside the **output** file “student_statistics.txt”:

Eddy Sheen

Highest grade: 97

Lowest grade: 63

Class average: 82.5

Freshmen: 3

Sophomores: 1

Juniors: 1

Seniors: 1

2. (35 points) Given an array of strings strs, group **the anagrams** together. You can return the answer in **any order**.

   An **Anagram** is a word or phrase formed by rearranging the letters of a different word or phrase, typically using all the original letters exactly once.

   Input type: A list with words;

   Output type: A list of lists with grouping the anagrams together

   Example 1:

   Input: strs = ["eat","tea","tan","ate","nat","bat"]

   Output: [["bat"],["nat","tan"],["ate","eat","tea"]]

   Example 2:

   Input: strs = [""]

   Output: [[""]]

   Example 3:

   Input: strs = ["a"]

   Output: [["a"]]

Note: **You cannot use any built-in functions. You can have only string, tuples, dictionaries to solve this problem. Follow the instructions that I have mentioned during the class**.

3. (30 points) Write the OOP program in python using class. Assuming you have four classes: Bank account which is the parent class and it has two child classes Saving Account class and Checking account class. Customer is another class who has a bank account; either saving or checking or both. Implement the scenario using python OOP and make sure you have covered those OOP concepts on your code: inheritance(any), polymorphism (runtime and compile time), abstraction and encapsulation.