Programming Assignments #4 and 5
CS 202 Programming Systems

Programming in Python
In Programs #4 and 5, you will be implementing your solutions using Python using VScode and providing a test suite compatible with Pytest. Your goal must be to develop a solution using abstractions (classes) but this time to implement it in Python. Make sure that your Design is not centered around your data structures – your data structures support the design but shouldn't be the primary emphasis of your design. Please make sure to implement these concepts:

- **Indicate with underscores your class instance variables (i.e., the fields) and these should not be accessed outside of an inheritance hierarchy**
- Develop an inheritance hierarchy; there must be a minimum of 5 total classes with 3 of them in a hierarchy. *These should not be isolated to just your data structures.*
- It is expected that you will use the NumPy library for arrays and make sure to also use Python lists.
- The application that USES the hierarchy must be in a class on its own.
- Implement at least one constructor with arguments
- Implement at least two functions using function overloading between classes and experiment with the way function overloading works in Python. ***Write about this.
- Implement at least two "operator" functions and experiment with how they work in Python (refer back to our lab on operator overloading).
- Try out the use of "super" in invoking a base class' methods.
- **Create a testing module to be used with Pytest.** Once your software is designed, then create the testing module(s) to perform black box testing. Then, updated it once you have implemented each method. With this, it should be clear to us that you have performed unit testing!
- *There is no limit to the number of files allowed for this assignment.*
- *We are looking for a comprehensive knowledge that classes exist to have a job to do. Classes with only setters and getters (or only input and display) will not count.*

For each of the above that you experiment with, write up information about it in your efficiency write-up. Instead of a debugger writeup, please discuss how you used VScode.

Program Requirements

How do you like using Canvas? For those of you who have spent most of your time with D2L, it is a big shift. For others, you may be enjoying its speed and flexibility. From the teaching aspect, there is much more that I can do to create a course with Canvas. However, I also think it is much harder for the students to navigate given the course content is not hierarchical. The biggest issue is the gradebook. None of us are notified when comments get posted in the gradebook. This is a "feature"!!

For this programming assignment, you will implement your own gradebook program. It can be modelled around our course or it can be based on one of your other courses.

Your gradebook needs to support three different kinds of grades. For example, they could be grades representing: Programming Assignments, Exams, or Proficiency Demos. Or, you could support other types of grades, such as practice quizzes, prelabs, completion of the linux & vim exercises each week, as well as attendance. Select grade items that are similar but yet different to support a hierarchical solution. **Please pick three (at most) of these, or others, to focus on for your gradebook. At least one of your selections should have a number of grades tied to it** (e.g., a list)…such as a programming assignment has a draft submission and two progress submissions along with a couple of writeups. Consider selecting grades that have different impact on the overall grade computation or score (numeric 0-100 versus alphabetic A-F or P/NP)

As usual, each class must have three "jobs" to do in addition to standard constructors, destructors, display and operator functions. This is your time to be serious about creating classes that have a clear purpose!

Your job is to create a program in two phases. You will begin with Program #4 and then conclude the assignment with Program #5. Think of it as something that would ultimately be a usable application for a University. Begin with creating a design which <u>must include</u> a plan for building a test suite using black box testing. <u>This is a required deliverable for the end of the first week.</u> The next page outlines the progression from Program #4 to Program #5.

Program #4 – Data structures

Step 1: Before the program can be used, you need to first set up what a single grade is – creating a hierarchy of classes where we push up the common and derive the differences for the three different types of grades. Each class must have three "jobs" to do in addition to what was outlined on the previous page! Really think about what the client would want to do with a grade. Remember, the operators do not count as having a "class" have a job to do. They are there in a support role.

When you select the three types of grades for this hierarchy, pick grade items that are similar but yet different – so that you can realize the benefit of inheritance.

Apply unit testing! Test each method individually for each class prior to progressing.

Step 2: Once you have a single grade completed, then create a collection of all of the grades for a given (one) student. *We expect a **linear linked list** to be implemented from scratch in this assignment using recursion.*

Program #5 – Then, once the data exists, with Program #5 we will create the actual collection of grades for all students in a class. With this part of the assignment, we want to be able to quickly look up a students' grades without performing a sequential search. To do this effectively, we will be implementing a binary search tree or balanced tree – which ever you did not implement in Program #3. Balanced trees do <u>not</u> need a remove function (so you should insert, search and display).

1. **Choice #1: Implement a Binary Search Tree**. It should have a full complement of insert, search (eg., find grade information for a particular student), display and remove functions.
2. **OR, Choice #2: Implement a BALANCED Tree** of your choice

CS202 - Checklist for First Week of Cycle

Cycle	Tuesday	Wednesday	Thursday	Friday
First week	Discussion (Begin Planning the classes)		Discussion Response	**Framework of your classes** **Black Box Testing Plan with specifics**
Second week	Discussion	Progress Submission: (prog 4)	Discussion Response	Finished Program 4 & Writeups (Efficiency and IDE)
Program #5 Finals Week				Thursday Finished Program 5 & Writeups (Efficiency and IDE)

______1 **Tuesday – Discuss your Plan**

- Using the Canvas Discussions; share ideas about:
  1 What three derived classes are you going to support
  2 What functions do you want for those three classes
  3 What information will be pushed up versus derived
  - **Make sure to ask a question to help others respond to you!**

______2 **Thursday – Constructive response to your Virtual Group**

______3 **Friday - Submit** Basic framework and Test plan

1. Provide the framework of a base class and three derived classes
2. Provide 3 functions per class supporting function overloading. These should be detailed and code recommended.
3. Include your plan for blackbox testing – be comprehensive on the cases that will be tested. Building a test suite ahead of time is what we are looking for!

- Due **by 7pm**. Submit to **Assignments** on Canvas

CS202 - Checklist for <u>Second</u> Week of Cycle

Cycle	Tuesday	Wednesday	Thursday	Friday
First week	Discussion (Begin Planning the classes)		Discussion Response	**Framework of the hierarchy and Test Plan for BlackBox testing**
Second week	Discussion	Progress Submission: (prog 4)	Discussion Response	Finished Program 4 & Writeups (Efficiency and IDE) with Pytest component
Program #5 Finals Week				Thursday of finals week Finished Program 5 & Writeups (Efficiency and IDE) with Pytest component

______1 **Tuesday – Discuss** with your virtual group

a. Discuss what changes you would make to your approach
b. Are you encountering problems?
c. How are you supporting recursion?

______2 **Wednesday - Submit** a Progress Submission

a. Due **by 7pm**
b. Submit to **Assignments** on Canvas
c. The remaining work at this point should be focused on cleaning up your code, developing application specifics, and resolving remaining bugs.

______3 **By Thursday – Constructive response to your Virtual Group**

a. Are there changes you would make to the approach for the next programming assignment?

______4 **Friday - Submit** a **Completed Assignment for Program #4 with Efficiency Writeup and Pytest test suite**

a. Due **by 7pm**
b. Submit to **Assignments** on Canvas
c. Upload the Efficiency and IDE Writeups separately

______5 **FINALS WEEK: Thursday - Submit** a **Completed Assignment for Program #5 with Efficiency Writeup with Pytest test suite**

a. Due **by 7pm** - In your final writeup evaluate VScode.