SE126 – 202420

Lab #5

In this lab, you will build a Python application that allows a user to repeatedly choose an option from a menu to search through the data provided in the text file: lab5.txt

**Part 1 – Build a Menu Function**

This function will return the user’s choice, based on the following options:

1. See All Student Report
2. Search for a Student [ID]
3. Search for a Student [Last]
4. View a Class Roster [class1, class2, and class3]
5. Exit/Quit Program

**Part 2 – Store File Data to 1D Parallel Lists**

Store the data from the provided file into respective lists. The file is set up as below:

ID LastName FirstName Class1 Class2 Class3

**Part 3 – Build Your Sequential Search**

Sequential search, while slow, allows for multiple records of matching data to be found.

Option 4 will require sequential search to show all students enrolled in a course. The user should be able to enter any casing of the class and have it found (if it exists)

Allow the user to accept a class name, and if the course exists: display all the students enrolled in this course (just student ID, first name, and last name). Use a new list (found\_class) to store which students are enrolled in the course and use this list to display the student information back to the user. If the class does not exist, tell the user this and allow them to revisit the menu to try again.

NOTE**:** you will need to search through ***all 3 class lists*** before your sequential search is complete.

**Part 4 – Build Your Binary Search**

Binary search requires an ordered list of *unique* values.

Options 2 & 3 will require binary search to be performed on either the student id value or the student last name value. The file is already in the proper order, so you shouldn’t need to bubble sort the lists. The user should be able to enter the Last name with any casing and have it found.

Allow the user to enter a searched-for ID or Last Name. Using binary search, display all of the student record data back to the user if the student they searched for is found. If the student is not found, tell the user this and allow them to revisit the menu to try again.

More on next page🡪

**Part 5 – Build the Final Application**

Build a Python application that repeatedly shows a user the menu and performs their choice until they choose to exit/quit.

Here is what should happen for each option of the menu:

1. See All Student Report
   1. Show all file data that has been stored to the lists, using the lists :]
2. Search for a Student [ID]
   1. Allow the user to enter an ID, and if that ID exists show the entire student record; alert the user if the ID does not exist
3. Search for a Student [Last]
   1. Allow the user to enter a last name, and if that name exists show the entire student record; alert user if the name does not exist.
   2. Accept any casing of the name
4. View a Class Roster [class1, class2, and class3]
   1. Allow the user to enter a class name, and if that class exists show the entire class roster: student IDs, First Names, and Last Names
   2. alert user if the name does not exist.
   3. Accept any casing of the class name
5. Exit/Quit Program
   1. Display a goodbye message when chosen

**Extra Credit – Build a Binary Search Function [+10 Points]**

For Options 2 & 3, use a binary search function.

This same function should be used for both options.

This function should handle the entire binary search: asking the user for the searched-for item, run the binary search algorithm, check to see if the searched-for item was found, and display appropriate information to the user based on the search results.

Submit your completed and documented final Python file to the assignment space by the end of day Wednesday of W8.