Sometimes a manager needs to look at historic data to make decisions about the future. A manager needs to be able to look at both aggregate and specific data about their data and make some decisions. The students have unique names. The costs of each student (like tuition, lunch amount owed, status cost, etc.) are tracked. Note that it is the actual amounts (for tuition, lunch, status, etc.) are stored and not the codes.

Write a program that processes student data to provide information about:

* A specific student and a specific amount that they owe for tuition or a specific fee (e.g., Bob Smith owes $200 for the sports fee).
* Aggregate information about a specific fee or tuition (e.g., the total tuition across all students or the total fees owed across all students playing sports)

Some general ideas about the program:

1. Your program must have a single dictionary to contain the data about the students and the tuition/fees (hint: you need a nested dictionary to do this)
2. Your dictionary must be loaded with at least four students (at a minimum you must include Bob Smith, Sue Jones, Mike Johnson, and Mary Thomas). Each student must have at least two different costs (tuition and/or fees) associated with them and at least one student must have three costs (at a minimum you must have tuition and lunch). Each fee or tuition will have the amount of the fee or tuition. For example, Bob Smith may have tuition of 8000 and sport fee of 200.

If a student doesn’t have a fee, then NO ENTRY should be put in the dictionary at all for that fee. For example, if Mary Thomas is not playing a sport, there should be no entry for sports in her record because her fee is 0 … it SHOULD NOT have an entry like sports:0). You may have more than the minimum if desired and time permits.

1. Your program must do two things as specified above:
   1. Aggregate information about tuition or fee action across all students (e.g., the total tuition across all students or the total lunch costs across all students)
      1. This part will ask the user what cost (tuition or fee) they wish to get aggregate information about.
      2. After receiving the input action, the program will then look through the dictionary and return a total of that cost across all students.
      3. You must use a function to process the dictionary to get the cost (hint: you’ll need to pass the cost that you are looking for as well as the dictionary to accomplish this)
      4. The output should look similar to the following: sports ----> 1000
   2. Provide specific student and the specific tuition or fee cost for that student
      1. This part will ask the user for a student and which tuition or fee cost that they wish to see (i.e., what’s the cost associated with that fee or tuition)
      2. After receiving the student and the tuition or fee, your program will then use the dictionary to see what is the cost for that tuition or fee that specific student.
      3. The output should look similar to the following: Bob Smith owes 8000 for tuition
      4. If a user is looking up tuition or cost and that student does not have that item (i.e., it is not present in the dictionary for that student), you should print a message for the user, instead of the output in 3biii that is similar to “Bob Smith doesn’t play sports” or “Bob Smith bring his own lunch” or “Bob Smith isn’t a new student or graduating.”
2. Note that the program should run correctly regardless of which student or cost (tuition or specific fee) is selected by the grader. The program should be effective in how it runs. For example, assume we were getting a total of four numbers from a list. It could be accomplished as A + B + C + D; however, having a loop to go through the list to do the adding would be more effective because the program could have millions of numbers and A + B + C + D … would not be feasible. While the dictionary will have a set number of items in it (based on how you load it), your logic should be able to work regardless of the number of students and/or costs (tuition or fees).
3. As a reminder, the following costs are associated with tuition and the various fees:
   * Tuition
     + In-district students pay $8,000 for annual tuition
     + Out-of-district students pay $9,500 for annual tuition
   * Sports
     + Students playing sports are charged an additional $200 sport fee
     + Students not playing a sport have no sport fee charged (i.e., $0.00)
   * Lunch
     + For students who bring their own lunch, there is no lunch fee (i.e., $0.00)
     + For students who get the standard lunch, the fee is $900 per year
     + For students who get the premium lunch, the fee is $1,170 per year
   * Student Status
     + Students in their initial year pay a “start-up” fee of $200 for the administrative setup to get them into the system, provide a locker, etc.
     + Students in their graduating year pay a graduation fee of $100 to cover the cost of the diploma, cap and gown rental, etc.
     + Students who are not in their initial year or graduation year have no status fee charged (i.e., $0.00)

**Grading Criteria**

|  |  |
| --- | --- |
| **Element** | **Points** |
| Program executes correctly and effectively | 20 |
| Proper dictionary including correct data inclusion | 15 |
| Proper use of function with return to get aggregate information about a cost (tuition or fees) across all students | 20 |
| Proper processing reporting of specific tuition/fee for specific student | 15 |
| Message for no tuition/fee | 15 |
| Comments | 15 |
| Total Points Possible | 100 |