# Lists and Loops Assignment

## List Manipulation – Random Student Selector

A screenshot of a computer program

Description automatically generated with medium confidence

### Description

Write a python program that randomly selects a student name from a list. It allows you to perform various actions related to selecting and managing student names. Here's a breakdown of what the program does:

1. It defines a function called **prompt()** that displays a menu of options for the user to choose from.
2. It defines a function called **select\_random\_name()** that selects a random student name from the **names** list. It checks if the list is empty or if all names have been used before returning a name.
3. It defines a function called **list\_names()** that displays the names in the **names** list along with their eligibility status (whether they have been used or not).
4. It defines a function called **update\_list()** that allows the user to enter a new list of names and updates the **names** and **used** lists accordingly.
5. It defines a function called **reset\_eligibility()** that resets the eligibility status of all names in the **used** list to **False**.
6. It defines a function called **run()** that repeatedly displays the prompt, reads the user's input, and performs the corresponding action based on the input.
7. The main program initializes the **names** and **used** lists with some initial values and then calls the **run()** function to start the program.

In summary, this program allows you to select a random student name, list the names along with their eligibility status, update the list of names, reset the eligibility status, and quit the program.

Sample Run

Here’s a sample session:

RANDOM STUDENT SELECTOR

---------------------

Press 'P' to pick a name

Press 'L' to list

Press 'U' to update the list

Press 'R' to reset eligibility

Press 'Q' to quit

Press any other key for a random name

===> L

Name Eligible

Bob True

Pete True

Mary True

RANDOM STUDENT SELECTOR

---------------------

Press 'P' to pick a name

Press 'L' to list

Press 'U' to update the list

Press 'R' to reset eligibility

Press 'Q' to quit

Press any other key for a random name

===> P

Selected name: Mary

RANDOM STUDENT SELECTOR

---------------------

Press 'P' to pick a name

Press 'L' to list

Press 'U' to update the list

Press 'R' to reset eligibility

Press 'Q' to quit

Press any other key for a random name

===> L

Name Eligible

Bob True

Pete True

Mary False

RANDOM STUDENT SELECTOR

---------------------

Press 'P' to pick a name

Press 'L' to list

Press 'U' to update the list

Press 'R' to reset eligibility

Press 'Q' to quit

Press any other key for a random name

===> U

Enter a space-separated list of names: bill tom jackson

RANDOM STUDENT SELECTOR

---------------------

Press 'P' to pick a name

Press 'L' to list

Press 'U' to update the list

Press 'R' to reset eligibility

Press 'Q' to quit

Press any other key for a random name

===> L

Name Eligible

bill True

tom True

jackson True

RANDOM STUDENT SELECTOR

---------------------

Press 'P' to pick a name

Press 'L' to list

Press 'U' to update the list

Press 'R' to reset eligibility

Press 'Q' to quit

Press any other key for a random name

===> P

Selected name: tom

RANDOM STUDENT SELECTOR

---------------------

Press 'P' to pick a name

Press 'L' to list

Press 'U' to update the list

Press 'R' to reset eligibility

Press 'Q' to quit

Press any other key for a random name

===> L

Name Eligible

bill True

tom False

jackson True

RANDOM STUDENT SELECTOR

---------------------

Press 'P' to pick a name

Press 'L' to list

Press 'U' to update the list

Press 'R' to reset eligibility

Press 'Q' to quit

Press any other key for a random name

===> R

RANDOM STUDENT SELECTOR

---------------------

Press 'P' to pick a name

Press 'L' to list

Press 'U' to update the list

Press 'R' to reset eligibility

Press 'Q' to quit

Press any other key for a random name

===> L

Name Eligible

bill True

tom True

jackson True

RANDOM STUDENT SELECTOR

---------------------

Press 'P' to pick a name

Press 'L' to list

Press 'U' to update the list

Press 'R' to reset eligibility

Press 'Q' to quit

Press any other key for a random name

===> Q

Process finished with exit code 0

### Rubric

|  |  |
| --- | --- |
| 15 | The main program initializes the **names** and **used** lists with some initial values and then calls the **run()** function to start the program. |
| 10 | Use the \* operator to initialize the **used** list |
| 10 | Initialize the list to [Bob, Pete, Mary]. You can hardcode this. |
| 10 | Create a ‘game loop’ in run() that uses the prompt() and calls the other functions accordingly. **run()** repeatedly displays the prompt, reads the user's input, and performs the corresponding action based on the input. Follow the formatting an options list illustrated above. Note that users can enter either uppercase or lowercase letters. Also note the last option, where any key (other than those with special functions) will pick a new name. This is considered the ‘power user’ interface. |
| 15 | **select\_random\_name()** selects a random student name from the **names** list. It checks if the list is empty or if all names have been used before returning a name. |
| 15 | **list\_names()** displays the names in the **names** list along with their eligibility status (whether they have been used or not). You must use a for-loop along with the built-in **enumerate()** method when iterating. Follow the formatting illustrated above. |
| 15 | **update\_list()** allows the user to enter a new list of names and updates the **names** and **used** lists accordingly. Reading a space-separated string from the command line is sufficient. |
| 10 | **reset\_eligibility()** resets the eligibility status of all names in the **used** list to **False**. |

### Submission

Provide your ***zipped PyCharm project*** to the assignment drop box.

Provide an ***expository video*** of a screen session where you describe each rubric point by stating the rubric point, demonstrating the functionality, and then DESCRIBING the code you wrote that implements the feature. Your video should be less than 5 minutes in length. Upload your video to the assignment drobox or provide a link to private, cloud-based video file. See the course materials for more info on expository videos, if needed.