**In-Class Practice 1**

**Empirical Methods for Innovation Strategies**

Name and ID: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

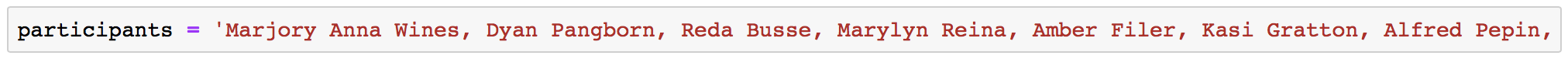
The exam is open-book, but not open-laptop!

1. **Translate into code**

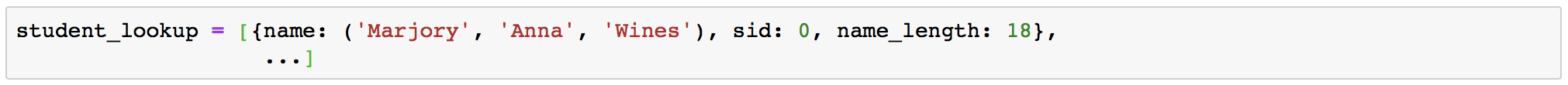
**Program 1 (9 points)**

1. create a list students from the string participants (1 point)
2. create a dictionary name2id that maps each student name to a unique ID (2 points)
3. create the inverse dictionary, id2name, which maps from the unique ID to the name (2 points)
4. create a new list student\_lookup (1 point)
5. iterate over the keys and values in name2id (1 point)
6. for each name and id pair, create a new dictionary. It should have the keys name (a tuple with all parts of the name, i.e., first, middle, last), sid (the unique integer ID from before), and name\_length (the length of the full name in characters). Add this dictionary to student\_lookup (2 points)

Input:



Output:



|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

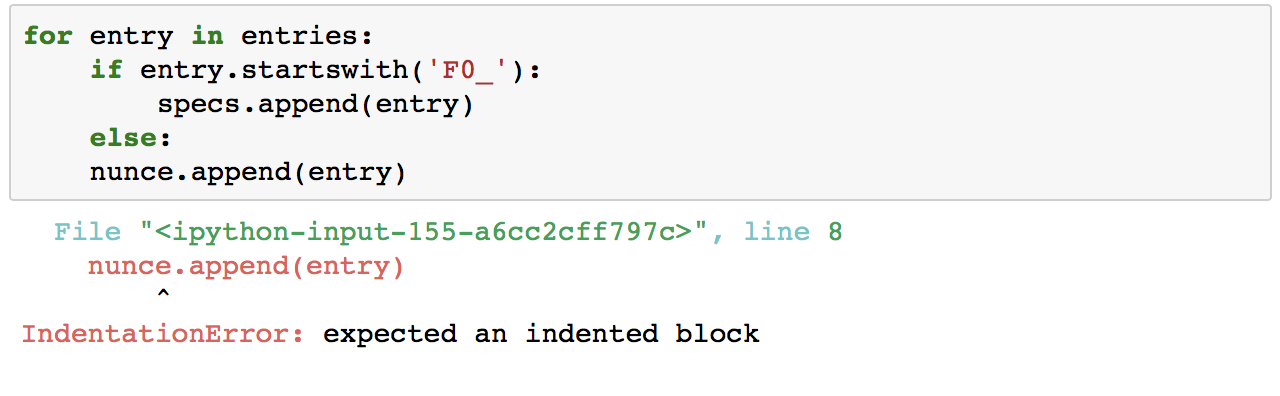
**Program 2 (7 points)**

You are given a list of strings called tasks, which contains duplicates. Create a new list importance\_order, which contains the items in reverse order, but each item only once.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

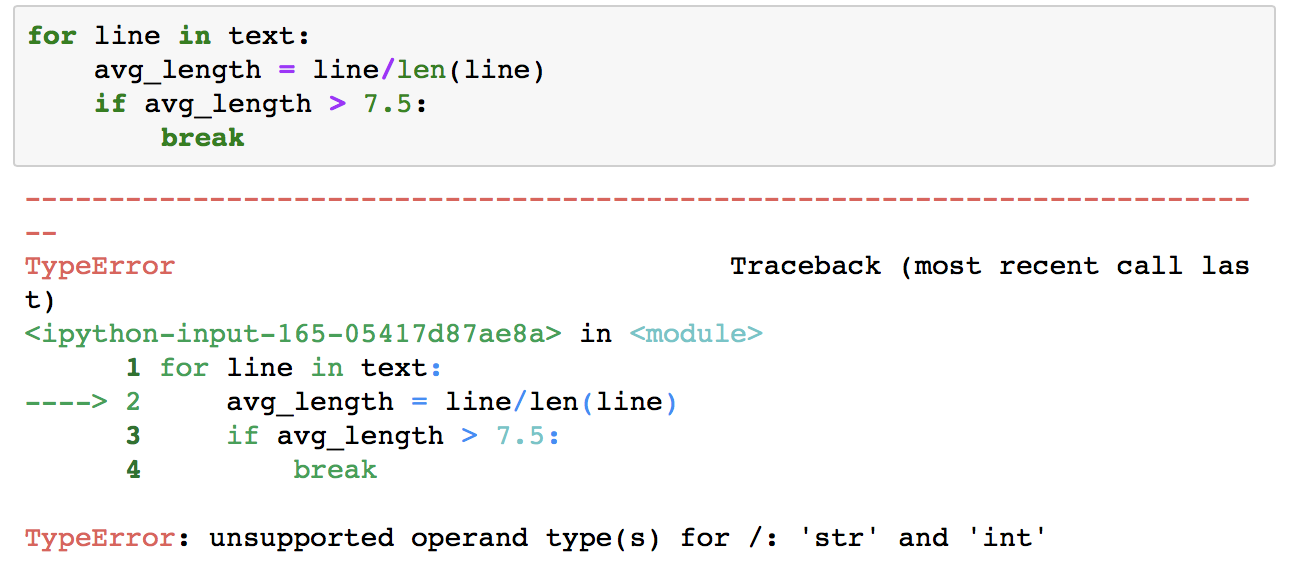
1. **Bloopers (4 points)**

This code contains something wrong (see the error message). Please explain what the mistake is and how to correct it ***(2 points)***.



|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

This code contains something wrong (see the error message). Please explain what the mistake is and how to correct it ***(2 points)***.



|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

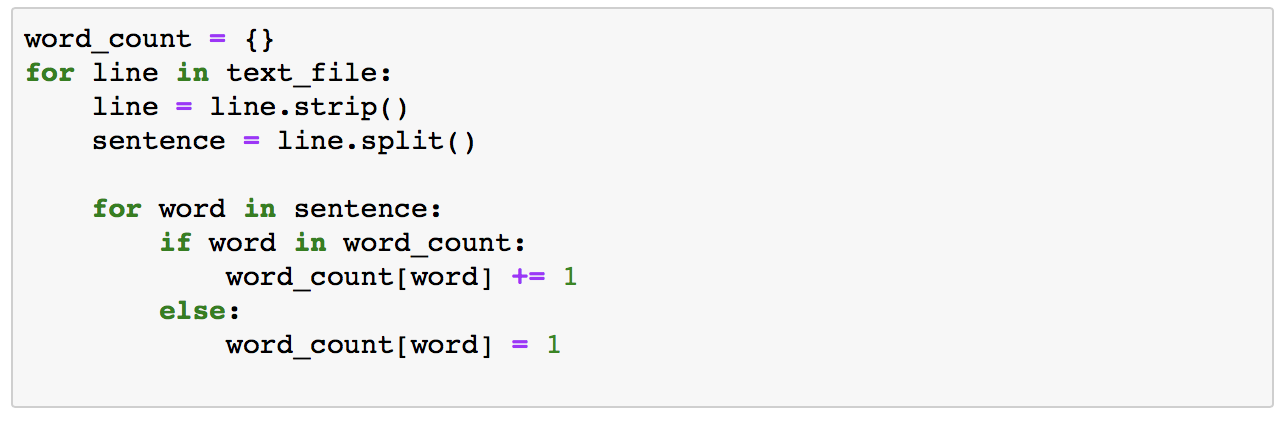
1. **Code Description (11 points)**

Describe what this code does (max 200 words) ***(5 points)***

|  |
| --- |
|  |
|  | |
|  | |
|  | |
|  | |
|  | |
|  | |
|  | |
|  | |
|  | |
|  | |
|  | |
|  | |
|  | |
|  | |
|  | |
|  | |
|  | |
|  | |
|  | |
|  | |
|  | |
|  | |
|  | |

Describe what this code does (max 200 words) ***(5 points)***

BONUS: Do you see something that could make the code simpler? ***(1 point)***



|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |