# A grey and white logo Description automatically generatedAssignment 3 – Python Lab

Objective : Use Python language and apply GenAI

Use copilot to complete the following exercises. Take screenshots of your code and submit that in word document. Submit **firstname\_assigment3.py** file as well.

Required Submissions: **firstname\_assignment3.py** file and **firstname\_assignment.doc**.

**Question 1**

A screenshot of a screen

Description automatically generated

1. Use the above table data. Create a dictionary. use for loop to display the results
2. Use the same data. Create a list, display the results.
3. Write your understanding of the difference between list and dictionary

**Question 2**

Use Copilot chat and perform the following .

1. open the csv file called "nfl\_offensive\_stats.csv" and read in the csv data from the file
2. Include the prompt and execute the code
3. the 3rd column in data is player position, the fourth column ③ is the player, and the 8th column is the passing yards. ③ For each player whose position in column 3 is "QB", determine the sum of yards from column
4. print the sum of the passing yards sorted by sum of passing yards in descending order

**Question 3**

1. John is signing up for a new social network website called ProgrammerBook. He wants to make sure that his password is strong.
2. John starts with a modest definition of what it means for a password to be strong: it’s strong if it’s not the word *password* and not the word *qwerty*. (Those are terrible passwords, for sure, but in reality, we have to do way better than this definition to ensure that our password is strong!) A helpful function would be one that takes a proposed password and tells us whether it is strong or not.
3. Unlike our previous functions in this chapter, we’re not dealing with numbers here. The parameter, the password to check, is text. And the return value is supposed to indicate some yes/no result. We need new types!
4. The Python type for text is called a *string*. There are zillions of possible strings because we can use a string to store whatever text we want. And the Python type for a yes/no result is called a *Boolean* or *bool*. A bool has only two values: True or False.

**Question 4**

Use prompt engineering technique . use the below to determining the largest number of students that can be added to a row. Explain your understanding .

**def most\_students(classroom):**

**'''**

**classroom is a list of lists**

**Each ' ' is an empty seat**

**Each 'S' is a student**

**How many new students can sit in a row?**

**'''**

① The improved prompt says we specifically want the maximum number of ' ' characters in any given row.

② count is a list function that returns the number of the argument in the list.

③ Code to keep track of maximum seats

Rubrics

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Evaluation criteria | **Not acceptable** | **Below** **Average** | **Average** | **Competent** | **Excellent** |
|  | **0% - 24%** | **25%-49%** | **50-69%** | **70%-83%** | **84%-100%** |
| Functionality | Missing all functionalities required | Some requirements are implemented. | Majority of requirements are implemented but some are malfunctioning. | Majority of requirements implemented. | All  requirements are implemented Correctly. |
| Documentation | No comments explaining code changes. | Minor comments are implemented and explained. | Some code changes are correctly  commented and explained . | Majority of code changes are correctly commented  and explained  . | All code changes are correctly commented  and explained  . |
| Testing & Evaluation | No evidence of testing and evaluation of the requirements. | Minor evaluation and testing efforts. | Some of the requirements have been tested & evaluated. | Majority of requirements are tested & evaluated. | Realistic evaluation and testing, comparing the solution to the requirements. |