

TOPIC 13

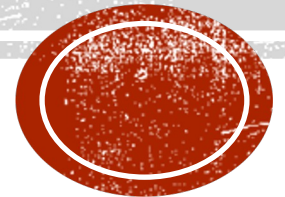
COURSE REVIEW

Summer 2022

Systems and Computer Engineering

Carleton University

Dr. Rami Sabouni



Objectives

- A project to review most of the topics covered throughout the semester
 - Topic01: Tuples
 - Topic 02: Import and modules
 - Topic 04: Dictionaries
 - Topic 06: Nested loops
 - Topic 08: Sorting Elements
 - Topic 09: User Interface
 - Topic 10: Curve fitting

Problem statement

- Create an interactive program that will read the student's information from a csv file then perform few operations on the data
- The following features are expected in the program:
 - `load_data()` Load the data from the csv file
 - `add_student()` Add a student's information
 - `remove_student()` Remove a student's information
 - Get a student's information (check next slides)
 - `count_passed()` Count number of students passed the course (grade > C-)
 - `plot_grades()` Plot grade distribution (histogram) and curve fit it

Problem statement

load_data ()

- Read the csv file and load the data in a dictionary that is formatted as shown below:

```
{125862624: {'First Name': 'John', 'Last Name': 'Snow', 'Grade': 71}}
{127280328: {'First Name': 'Tony', 'Last Name': 'Stark', 'Grade': 72}}
{131239350: {'First Name': 'Bruce', 'Last Name': 'Wayne', 'Grade': 67}}
{123965110: {'First Name': 'Bruce', 'Last Name': 'Banner', 'Grade': 58}}
{126378590: {'First Name': 'Albus', 'Last Name': 'Dumbledore', 'Grade': 82}}
{126024463: {'First Name': 'Harry', 'Last Name': 'Potter', 'Grade': 75}}
{124516566: {'First Name': 'Hermione', 'Last Name': 'Granger', 'Grade': 57}}
{127094040: {'First Name': 'Tom', 'Last Name': 'Riddle', 'Grade': 68}}
{131377214: {'First Name': 'Peter', 'Last Name': 'Parker', 'Grade': 79}}
{124267150: {'First Name': 'Super', 'Last Name': 'Mario', 'Grade': 66}}
{132813544: {'First Name': 'King', 'Last Name': 'Kong', 'Grade': 77}}
{126729683: {'First Name': 'Guido', 'Last Name': 'van Rossum', 'Grade': 65}}
{123678146: {'First Name': 'Konrad', 'Last Name': 'Zuse', 'Grade': 62}}
{127435849: {'First Name': 'Alan', 'Last Name': 'Turing', 'Grade': 98}}
{130645143: {'First Name': 'Dennis', 'Last Name': 'Ritchie', 'Grade': 51}}
```

Problem statement

`add_student()`

- The function adds a student to the dictionary and verifies that the student has been added
- The function returns the updated dictionary
- The function prints a message stating:
 - The student has been added correctly
 - There was an error adding the student
- The function takes two arguments
 - the dictionary where the student must be added
 - a tuple argument that has:
 - student number
 - first name
 - last name and grade

Problem statement

`remove_student()`

- The function removes a student to the dictionary and verifies that the student has been removed
- The function takes two arguments:
 - the dictionary from where the student must be removed
 - the student number that needs to be removed
- The function returns the updated dictionary
- The function prints a message stating:
 - The student has been added correctly
 - There was an error adding the student

Problem statement

Get a student's information

- When getting student's information, the data returned should be sorted in ascending order by
 - `get_students_by_number ()` Student Number (Bubble Sort)
 - `get_students_by_first ()` First Name (Insertion Sort)
 - `get_students_by_last ()` Last Name (Selection Sort)
 - `get_students_by_grade ()` Grade (Merge Sort)

Problem statement

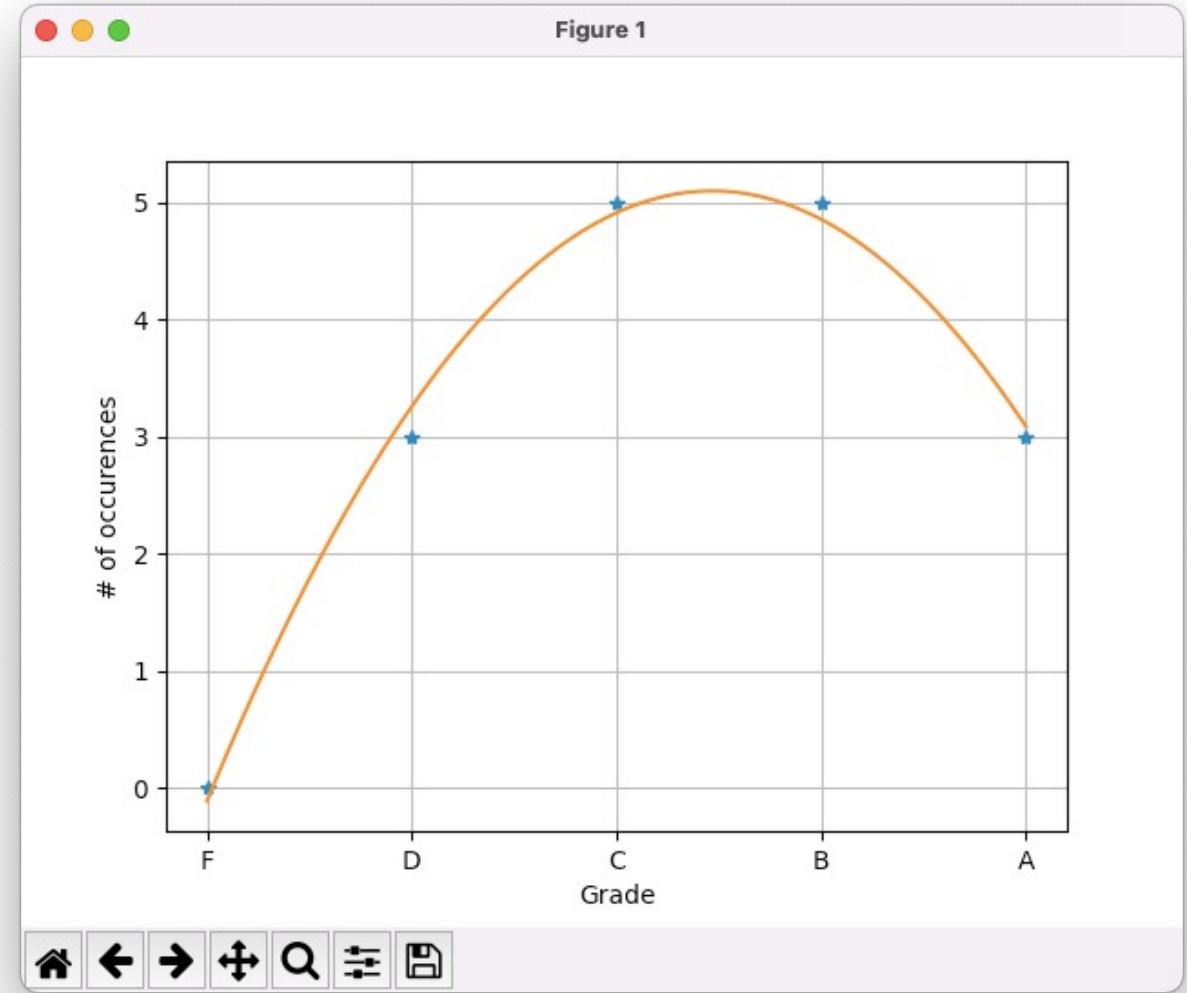
`count_passed()`

- Returns number of students in the course and number of students passed (over C-) as a tuple
- The function takes the dictionary from where the students grades are stored

Problem statement

`plot_grades ()`

- Plots the histogram of student's grades (A, B, C, D, F)
- Features used:
 - Dictionary to create the histogram
 - `polyfit()` for Curve fitting
 - `polyval()` for finding the y value based on the x value
 - plugging in the x values in the function from the curve fitting
 - Some extra plot manipulation settings



Problem statement

User Interface

- Create a user interface that can accept one of 7 commands as shown on the right
- The file has to be loaded from the csv file first
- If the user enters “G” they get to choose how will the retrieved data be sorted

```
1- L)oad file
2- A)dd student
3- R)emove student
4- G)et students (sorted)
5- C)ount number of students passed
6- P)lot grade distribution
7- Command line Q)uit
```

```
Please type your command: l
Enter file name:
Student_info.csv
```

```
1- L)oad file
2- A)dd student
3- R)emove student
4- G)et students (sorted)
5- C)ount number of students passed
6- P)lot grade distribution
7- Command line Q)uit
```

```
Please type your command: g
Select how to sort the retrived information:
_____ S)tudent Number _____ F)irst Name _____ L)ast name _____ G)rade : s
```



Questions?