

CSCE 110: Programming I Project

Fall 2020

The Grade Center

Description of the project: Python is a powerful tool for conducting analysis on large data sets and visualizing various attributes of those data sets. The objective of this project is to give you insight into this aspect of programming using Python. You will write a program that reads a data set in CSV format and analyzes the data set. You are allowed to use any libraries of your choice for this project.

The CSCE 110 instructor needs a program that reads a [CSV file of grades](#) and prints some statistics and charts from the grades.

Your program should repeatedly print the main menu that gives the instructor a list of options. To choose an option, the instructor should type the option number.

```
*****Main Menu*****
1. Read CSV file of grades
2. Generate student report file
3. Generate student report charts
4. Generate class report file
5. Generate class report charts
6. Quit
*****
```

Option 1: The instructor should be able to enter the path of a CSV file of grade. The program should read the file. All grades are on a scale of 100. **No other option should read a file.**

Option 2: The instructor should be able to generate a report for a student in a text file by entering the student's UIN. Name the file UIN.txt where UIN is the student's UIN. The file format is provided in the [Data](#) directory. The report should include the following:

1. Exams mean
2. Labs mean
3. Quizzes mean
4. Reading activities mean
5. Score as a percentage
6. Letter grade

Use the following table for your calculations

Grading Distribution	Standard Letter Grading Scale
Exam 1: 15%	A: 90 - 100

Exam 2: 15%	B: 80 - 89
Exam 3: 15%	C: 70 - 79
Labs: 25%	D: 60 - 69
Quizzes 10%	F: < 60
Reading activities: 10%	
Project: 10%	

UIN validation:

The UIN should be ten digits and should contain numbers only. If the UIN is not in the right format, the program should print an error message and ask again.

If the UIN is not found in the grade center, the program should print an error message and ask again.

Option 3: The instructor should be able to generate report charts for a student by entering the student's UIN. The program should save the charts in a directory for the student. The directory name should be the UIN. The directory should be located in the current working directory of the program. In the directory, include the following charts in separate png files. All charts must be annotated correctly with labels and a title.

1. Bar chart of exams
2. Bar chart of labs
3. Bar chart of quizzes
4. Bar chart of reading activities

Option 4: The instructor should be able to generate a report for the entire class in a text file report.txt. The file format is provided in the [Data](#) directory. The report should include the following:

1. Number of students
2. Minimum score
3. Maximum score
4. Median score
5. Mean score
6. Standard deviation

Option 5: The instructor should be able to generate report charts for the class. The program should save the charts in a directory named `class\_charts`. In the directory, include the following charts in separate png files. All charts must be annotated correctly with labels and a title.

1. Pie chart of the letter grades distribution
2. Bar chart of the letter grades distribution

Option 6: The instructor should type q or quit to quit the program.

Deliverables:

1. **The code:** Document your code properly. Every function must have a documentation string.
2. **The project documentation:** Generate the API documentation for your source code. Use pdoc: <https://pdoc3.github.io/pdoc/> to generate the documentation for your project. All the code must be documented properly. Here is an [example](#) of pdoc.

Demo video: <https://youtu.be/YJZ1W6ptjI>

Data and sample output:

https://drive.google.com/drive/folders/1bRES5h0HLQIPAA2tGfF_QSDcsU7y_T6L?usp=sharing

INSTRUCTIONS:**1. Project Submission Guidelines:**

The last day to submit your project is **Tuesday, December 08th, 2020, at 11:59 PM CST.**

You may resubmit your project as many times as you need until the due date. Only the most recent submission is graded. Submit all your files on [Gradescope](#).

Both team members must add their names to the same project in Gradescope.

No late submissions will be accepted.

The project submission should include:

- a. Your generated documentation
- b. Your project file(s)

You are required to include the following lines in the header of all your code files:

File: filename.py

Author: Team members names

Date: 11/23/2020

Email: Team members emails

Extra credit (5 points):

Rewrite your project code using object-oriented programming. Create a class Student and a class Classroom. The Student class must have its methods. The Classroom class must have its methods. You must put all the operations from options 1 to 5 in an appropriate class method.

Examples:

- The operation **Generate student report file** should be in the Class Student in a method named e.g. `generate_report()`.
- The operation **Bar chart of the letter grades distribution** should be in the Class Classroom in a method named e.g. `plot_grades_barchart()`.

These methods should be called using an instance of the Student class or the Classroom class.

Important note:

Your extra credit program must be a separate program. If you are submitting your project in Gradescope, make sure to separate the extra credit program from the original assignment program. We must be able to grade both programs independently.