

# Lab 11: File Lab-Grade Calculator

## Overview

In this lab, students will practice working with files. The created program will read from several data files, and calculate course statistics. This will require both reading files and iterating over a directory. Students will also be introduced to the [matplotlib](#) library.

## Specification

### Intro

Canvas has crashed once again, and we need a new program to calculate grades. However, all we have left of the course is [a folder with a few files in it](#). We don't know the format of these files, so you will have to read them and figure it out yourself.

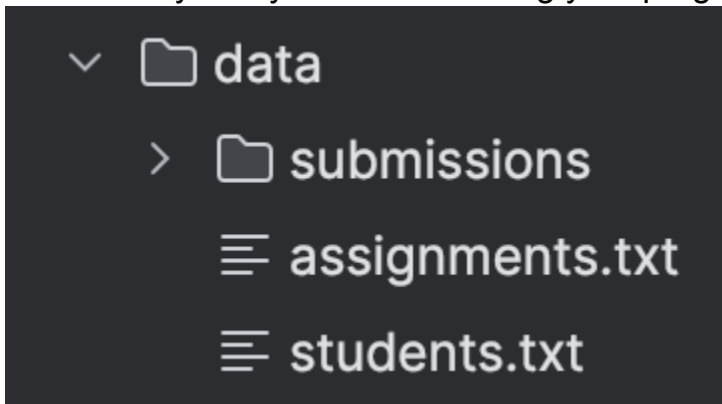
Here's everything we remember about the course data:

- Every student has a name and a 3 digit id number.
- Every assignment has a name, a point value, and a 5 digit id number.
- Every submission has a student id, an assignment id, and a percentage of the points earned for the assignment.
- The total number of points for all the assignments is 1000.
- Every student submitted every assignment, there were no missing submissions.

**Before you start working on the implementation, look at the provided files. Figure out how the files are formatted, and think about what concepts you have learned in this course that might help you organize the data.**

### Setup

To get started, [download the course data here](#). Unzip the file, and place the data directory in the directory that you will be running your program in. It should look something like this:



As an example, your program should access the students.txt file with the path 'data/students.txt'

# Lab 11: File Lab-Grade Calculator

Finally, install the library 'matplotlib' the same way that you installed pygame. This is a library for data analysis and visualization. We will be using it in part 3 to generate a histogram.

## Menu

When your program is run, it should print a menu and prompt the user for a selection. For this assignment, you do not need to worry about users entering invalid input. After the menu is printed and the selected task is run, the program should exit.

1. Student grade
2. Assignment statistics
3. Assignment graph

Enter your selection:

## Option 1

For option 1, the program should ask the user for a student name. It should calculate that student's grade for the entire course, or print "Student not found" if the student does not exist. Their grade should then be printed out as a percentage, rounded to the nearest whole percentage.

## Option 2

For option 2, the program should prompt the user for an assignment name. It should find the maximum, average, and minimum percent scores for all submissions for that assignment. If the assignment doesn't exist, print "Assignment not found".

## Option 3

For option 3, the program should prompt the user for an assignment name. If the assignment does not exist, print "Assignment not found". If the assignment does exist, the program should display a histogram of the assignment scores. We will use a library called [matplotlib](#) to create the graph. Check the example below, and play with the bins argument to make the graph more understandable. The chosen bins won't be graded, only whether or not your graph displays all the scores. **Take a screenshot of one of your graphs and submit it with your program file.**

```
# top of your program
import matplotlib.pyplot as plt

# Wherever you want to display the graph
plt.hist(scores, bins=[0,25,50,75,100])
plt.show()
```

# Lab 11: File Lab-Grade Calculator

## ↔ Examples

---

Here are examples for each of the menu options. **User input is in bold.**

### Example 1

1. Student grade
2. Assignment statistics
3. Assignment graph

Enter your selection: **1**

What is the student's name: **Michael Potter**

67%

### Example 2

1. Student grade
2. Assignment statistics
3. Assignment graph

Enter your selection: **2**

What is the assignment name: **Quiz 1**

Min: 56%

Avg: 72%

Max: 86%

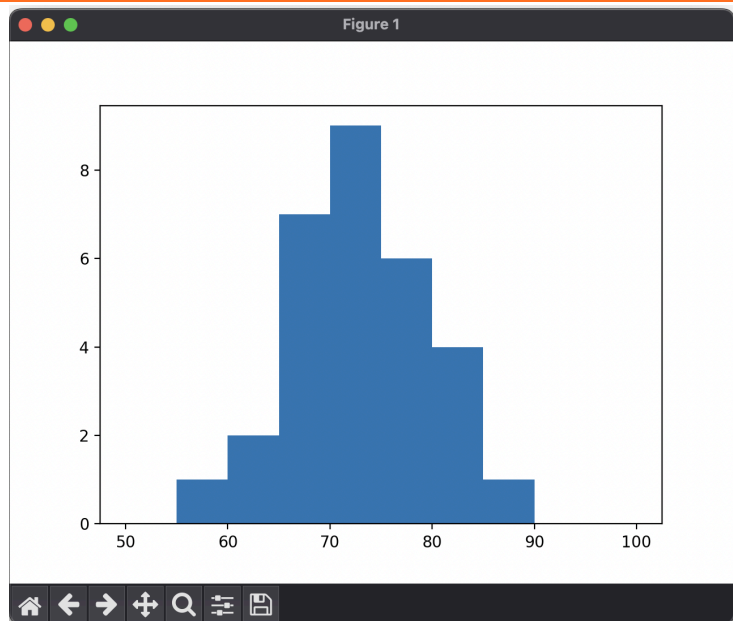
### Example 3 (This example will open a new window with a graph).

1. Student grade
2. Assignment statistics
3. Assignment graph

Enter your selection: **3**

What is the assignment name: **Quiz 1**

# Lab 11: File Lab-Grade Calculator



## Submission

To submit this assignment, create a new GitHub repository and commit + push your code.

**MAKE SURE YOUR IMAGE AND Lab11.py FILE ARE INCLUDED IN THE REPOSITORY WHEN SUBMITTING.**

You will first need to connect to GitHub, by making sure to authorize gradescope:

Connect your Account

[Connect to GitHub](#)

Repository \*

Select a repository...

Branch \*

Select a branch...

Cancel

Upload

Cancel

Authorize gradescope

Authorizing will redirect to  
<https://www.gradescope.com>

Select the proper repository (keep private!)

# Lab 11: File Lab-Grade Calculator

Repository \*

Select a repository...

newmanhw/gradecalculator

Last updated at: Nov 18, 2024 at 11:16 pm

Select main branch:

Branch \*

Select a branch...

main

This uploads everything in your repository to Gradescope submission.

Again, **MAKE SURE YOUR IMAGE AND Lab11.py FILE ARE INCLUDED IN THE REPOSITORY WHEN SUBMITTING.** We will be MANUALLY grading the image, so the autograded score maximum is 80/80.