# CSCI 127: Joy and Beauty of Data

Lecture 10: Files

Reese Pearsall
Snowmester 2020
https://reesep.github.io/classes/127/main.html

# Announcements

Lab 5 due **tomorrow** (Tuesday 11:59 PM)

- After today, you will be able to finish it
- Very similar to an example we will do later today

- There is an overview video posted

# Announcements

Program 2 due Wednesday @ 11:59 PM

- Should have everything you need to complete it
- I’ll likely make Wednesday’s lecture shorter to give you more time to work on it

# Announcements

Lab 6 due Thursday @ 11:59 PM

You have two options:

1. Come to a live virtual review session Thursday @ 11:00 AM in my Webex room
   - If you show up, you will receive a 100% for Lab 6

2. Answer any two example test questions and submit your python solution to D2L
   - This will largely be effort based… if you attempted them you will receive 100% for Lab 6

I’m hoping this will be an **easy**, **non-stressful** 100% for everyone…..

# Announcements

**Midterm Exam on Friday**

Will be posted at 6 AM on Friday. Due at 11:59 PM on Friday

I'm hoping that it shouldn't take more than 2 hours to complete

I'm still trying to figure out how I want it to be submitted (.py file, written down and scanned, etc)

**100 Points**

10 multiple choice Qs **(20 pts)** > Two of these will be pulled from the practice exam

3 short programming problems **(60 pts)**

1. Basic Python Calculation (20 pts)
2. Selection (30 pts)
3. Nested List (30 pts)

> One of these questions will be directly pulled from the example test questions

# Announcements

**Midterm Exam on Friday**

Content covered on midterm

Data Types/Variables
Operators (+,-,*,%,…)
User input
.format
Functions
Selection
Iteration (for loops/while loops)
Lists
Strings
Modules
Files

Good ways to study:

- Review Labs/Programs
- Review in class examples
- Do example test questions
- Read through textbook

# Announcements

Extra Credit Opportunity

If you fill out a short, anonymous mid-semester survey (coming soon), I will add 2% to your midterm grade.

Submit a screenshot that you completed the survey to D2L

** If you get 100% on the midterm, the 2% may or may not be added

More formal announcement and instructions coming very soon

# Announcements

You can now sign up for a 1 on 1 meeting time w/ Reese

Remember that **5%** of your final grade comes from meeting with me once during the semester

https://doodle.com/poll/civq435b9vgiy62w?utm_source=poll&utm_medium=link

- **EVERYONE** must sign up for a time via doodle. If none of the available times work, email me
- First come, first serve
- Please only sign up for 1 time slot. If you need more than 15 minutes, then feel free to book two consecutive time slots (times are in Mountain/Denver)

# Announcements

What you can expect from me

- I will have all labs/programs graded by the midterm exam (maybe not Program 2…)
- I will have solution videos for all assignments so far by the midterm
- I will post all example test questions tonight/tomorrow
- I will post practice exams tomorrow/Wednesday morning

- I will post lab 6, lab 7 and program 3  tomorrow

# Today:

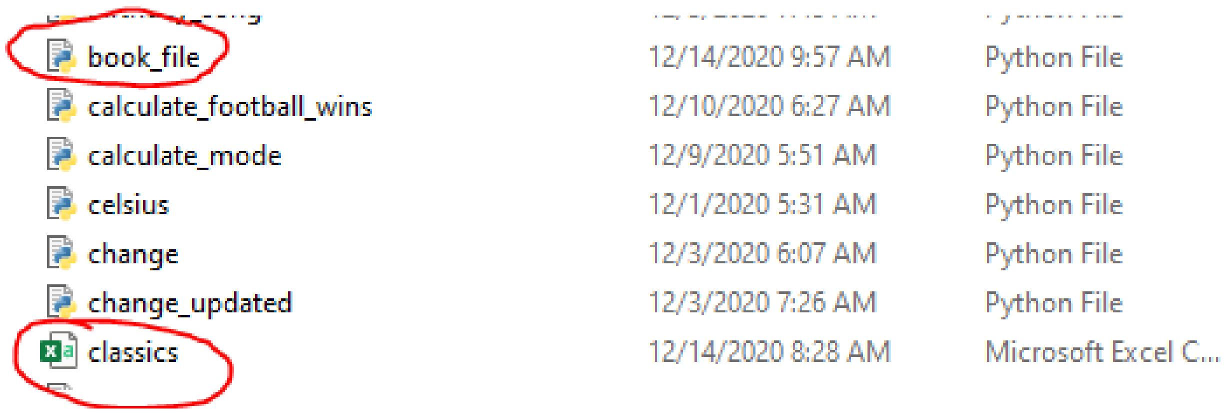
Dear Professor,
I hope you're doing well
your course has ruined my life and made me question my will to live

Reading from Files, Recursion (maybe?)

# Reading from files

## Steps for reading a file

**STEP 0: Make sure your .py file and file you are reading from are in the same folder/directory !!**



| | | |
|---|---|---|
| book_file | 12/14/2020 9:57 AM | Python File |
| calculate_football_wins | 12/10/2020 6:27 AM | Python File |
| calculate_mode | 12/9/2020 5:51 AM | Python File |
| celsius | 12/1/2020 5:31 AM | Python File |
| change | 12/3/2020 6:07 AM | Python File |
| change_updated | 12/3/2020 7:26 AM | Python File |
| classics | 12/14/2020 8:28 AM | Microsoft Excel C... |

# Reading from files

## Steps for reading a file

Name of file. Could be a .txt, .in, etc

“r” means you reading from the file

```
file = open(“my_file.csv”, ”r”)

file.readline()
for each_line in file:
    row = each_line.split(“,”)

    ## do stuff here with row


file.close()
```

1. Open the file

2. Discard the column headers (if necessary)

3. Do a for loop through each line in the file

4. Convert each line of the file (string) to a list using **.split()**. If you reading from a .csv file, you will always split on a comma (“,”), otherwise you will need to figure out what to split on.

Close the file

# Announcements (Tuesday):

Java
JS
Arrays start at 0
Lua
MATLAB
Arrays start at 1
Perl
Arrays can start wherever ¯_(ツ)_/¯
STAR WARS
Arrays start at 4, stop at 6, restart at 1, stop again at 3, restart at 7 then continue on

Lab 5 due **tonight** @ 11:59 PM

Program 2 due **tomorrow** @ 11:59 PM

Lab 6 due **Thursday** @ 11:59 PM**

Midterm Exam on Friday

Extra Credit Opportunity Available

You can now schedule your one on one meeting w/ Reese (5% of your grade)

Today: More on reading from files, writing to files

# Writing to Files

Name of file that will be created

“w” for write

```
file = open(“out.txt”, ”w”)
```

1. Open the file

```
for i in range(10):
    file.write(“hello x” +str(i))
```

2. Write to file using **.write()**
Note: You cannot use commas like you can with print()
Note2: Everything you write must be a string

```
file.close()
```

3. Close the file
**IMPORTANT**

# Announcements (Wednesday):

Program 2 due **TONIGHT** @ 11:59 PM

Midterm Exam on Friday
- Example Test Questions have been posted
- Extra Credit Opportunity posted
- Practice Exam(s) should be posted tonight/tomorrow

Lab 6 due tomorrow @ 11:59 PM
- Review session at 11:00 AM on Thursday

Slight changes to the schedule
- Short lecture today
- No (required) lecture to watch tomorrow and Friday

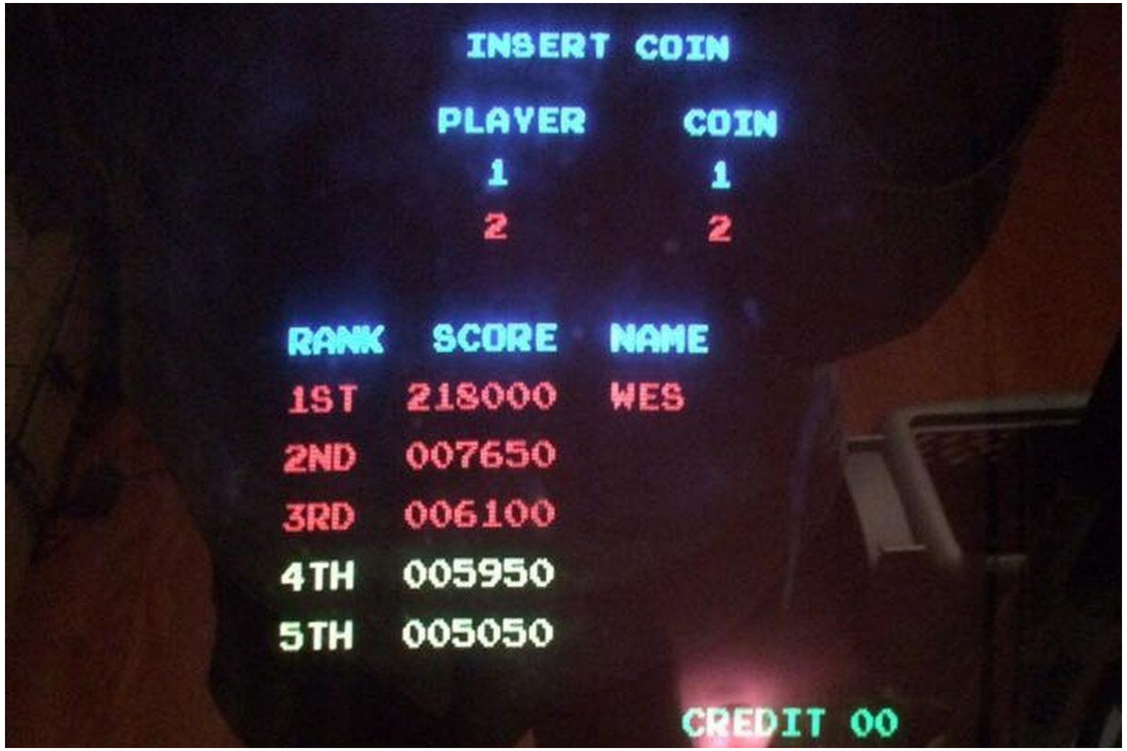
GroupMe
- Let me know if you need an invite
- Not allowed to access GroupMe during the Midterm
- Please be careful about oversharing solutions

# Example: Arcade High Score Table

Write a Python Program that will read in a series of 10 arcade game scores along with a player name. These scores are sorted from greatest to least

When the program is run, it should prompt the user for a new score and name. If the score the user inputs is the highest score so far, it should get placed at the very top of the scores. Otherwise, it should get placed in the appropriate spot in the top 10 (if it made it within the top 10)

The program should then write out to the same file it read from with the updated high score list. The high score list should ONLY have the top 10 scores and player names



INSERT COIN

| PLAYER | COIN |
|---|---|
| 1 | 1 |
| 2 | 2 |

| RANK | SCORE | NAME |
|---|---|---|
| 1ST | 218000 | WES |
| 2ND | 007650 | |
| 3RD | 006100 | |
| 4TH | 005950 | |
| 5TH | 005050 | |

CREDIT 00