

# Final Project

## Learning Objective

Demonstrate knowledge of Python programming concepts learned throughout the semester. You are not allowed to use Python concepts that were not taught this semester in this course.

## Assignment Description

Write a program that allows the user to learn about roller coasters. Read a CSV file with information about roller coasters to create a list of dictionaries. Each dictionary holds the information for one roller coaster. The user will be allowed to get information regarding the roller coasters including the amusement parks they are located in.

You are provided a CSV with data about roller coasters.

- Each row represents one line in a table, and commas separate each column.
- The first line in the CSV file represents the header that contains the keys for the dictionaries. Each subsequent row represents a coaster.
- The data may not be complete for each row.

### CSV File (Example)

```
name,material type,seating type,speed,height,length,num inversions,manufacturer,park,status
10 Inversion Roller Coaster,Steel,Sit Down,72,30,850,10,Intamin,Chimelong Paradise,status.operating
Abandon Mine,Steel,Sit Down,,,0,Molina & Son's,Uncle Bernie's Theme Park,status.operating
GhostRider,Wooden,Sit Down,90,36,1382,0,GCI,Knott's Berry Farm,status.operating
Zyclone,Steel,Sit Down,,,0,na,Funland Amusement Park,status.closed.definitely
Zydeco Scream,Steel,Sit Down,75,36,267,3,Vekoma,Six Flags New Orleans,status.closed.definitely
```

### CSV Data in a Table (Example)

name	material type	seating type	speed	height	length	num inversions	manufacturer	park	status
10 Inversion Roller Coaster	Steel	Sit Down	72	30	850	10	Intamin	Chimelong Paradise	status.operating
Abandon Mine	Steel	Sit Down				0	Molina & Son's	Uncle Bernie's Theme Park	status.operating
GhostRider	Wooden	Sit Down	90	36	1382	0	GCI	Knott's Berry Farm	status.operating
Zyclone	Steel	Sit Down				0	na	Funland Amusement	status.closed.definitely
Zydeco Scream	Steel	Sit Down	75	36	267	3	Vekoma	Six Flags New Orleans	status.closed.definitely

Your solution must use techniques from course materials. You may NOT import and use the CSV or other modules to process the CSV file.

## Steps

1. In PyCharm (Community Edition), open your existing ITP115 project.
2. Under the Assignments directory, create a new directory called **project\_last\_first** where *last* is your last/family name and *first* is your preferred first name. Use all lowercase letters.
3. In the directory, you will create **multiple Python files**. At the top of **each file**, put comments in the following format and replace the name, email, section, and filename with your actual information:  

```
# Name, USC email
# ITP 115, Fall 2022
# Section: number or nickname
# Final Project
# filename.py
# Description:
# Describe what this file does.
```
4. Put the **roller\_coasters.csv** file in your **project\_last\_first** directory.
  - Download the file from Blackboard under the item for the Final Project.
  - Drag it onto your **project\_last\_first** directory in PyCharm.
5. Create a Python file entitled **helper.py**. In this file, you will define functions that will be called from other Python files. Make sure to add the comment block at the top of this file.
  - Define the **createCoastersFromFile()** function.
    - Parameter: `filenameStr` is the name of the CSV file to read and set a default value of "roller\_coasters.csv"
    - Return value: a list of dictionaries where the keys are the strings from the header row and the values are strings containing the information from the rest of the CSV file
    - Each coaster is represented with a dictionary. The keys are the text from the header row, which are the following strings: "name", "material type", "seating type", "speed", "height", "length", "num inversions",

"manufacturer", "park", and "status". Get the keys from the file using the `file.readline()` method. Make sure to remove the new line ("`\n`") at the end of the line. Then create a list containing the keys using the appropriate string method.

- Each row for the remaining rows of the CSV file has information for a roller coaster. Make sure to get rid of the new line at end of each line by using the appropriate string method.
  - The status string for a coaster starts with "status." Do not include "status." in the value for the dictionary. One way to accomplish this is to use branching and slicing.
  - This function will have the following structure: Create a variable to hold a list of coasters. Open a file for reading, which creates a file variable. Read the keys in the first line. Loop through the rest of the file. In this outer loop, create a dictionary and get the data. Use a nested (or inner) loop to put the data in the dictionary, but make sure to handle the status correctly. After the nested loop, add the dictionary to the list. After the outer loop, close the file and return the list.
- Define the **`getParksList()`** function.
    - Parameter: `coastersList` is a list of dictionaries where each dictionary represents a roller coaster
    - Return value: a list containing the names of unique parks (strings) in alphabetical order
    - This function will have the following structure: Create a variable to hold a list of park names. Loop through the list of coasters. Get the park name from the dictionary and add it to the list if it is not already in the list. After the loop, sort the list and then return the list.
  - Define the **`getFastestCoaster()`** function.
    - Parameter: `coastersList` is a list of dictionaries where each dictionary represents a roller coaster
    - Return value: a dictionary containing one coaster, which has the largest value for speed

- Determine the fastest coaster by using the value of the "speed" key in the dictionary for each park. The value should be a string and may be an empty string. Before converting to an integer, make sure the string contains only digits.
  - This function will have the following structure: Create variables to hold the fastest speed and a dictionary. Loop through the list of coasters. Get the speed of the coaster and compare it to the appropriate variable. Reset the variables accordingly. After the loop, return a dictionary with the largest speed.
6. Create a Python file entitled **user\_interface.py**. In this file, you will define functions that interface with the user. These functions will be called in the main() function in the main.py file. Make sure to add the comment block at the top of this file.
- Import the **helper** file in order for functions in this file to be able to call functions in the helper.py file that you have defined.
  - Create the **displayMenu()** function.
    - Parameter: menuDict is a dictionary for the menu
    - Return value: None
    - Display the menu to the user using the menuDict parameter. The keys contain letters (strings) for the menu, while the values contain short descriptions (strings) for each letter. Print the key (letter) followed by "->" and then the value (description).
    - This function will have the following structure: Get the keys and sort them. Loop through the sorted keys and print the key, "->", and description. Since you have not created the menuDict yet, you will not know what it will look like.
  - Create the **getUserChoice()** function.
    - Parameter: menuDict is a dictionary for the menu
    - Return value: a string that is a valid choice entered by the user
    - Get input from the user using the following prompt:

<b>Choice:</b>
----------------

- Use the appropriate loop to continue to ask the user for input until they enter valid input. Allow the user to enter upper or lower case strings. Also allow the user to enter extra spaces before and after their input. The keys in the menuDict parameter have the valid letters (strings) which are upper case. Here is an example with user input in green text ("Q" is one of the keys):

```
Choice: AAA
Choice: bee
Choice: J
Choice: 1
Choice: x
Choice: q
```

- Create the **displayNumCoasters()** function.
  - Parameter: coastersList is a list of dictionaries where each dictionary represents a roller coaster
  - Return value: None
  - Display a message to the user with the total number of coasters in the list.
  - Here is an example:

```
The total number of coasters is 2796
```

- Create the **displayNumOperatingCoasters()** function.
  - Parameter: coastersList is a list of dictionaries where each dictionary represents a roller coaster
  - Return value: None
  - Display a message to the user with the total number of coasters in the list that are operating. Use the "status" key in the dictionary for each coaster.
  - Here is an example:

```
The total number of operating coasters is 2169
```

- Create the **displayCoaster()** function.
  - Parameter: coasterDict is a dictionary holding the data for one coaster
  - Return value: None
  - Display the information for the coaster in the following format:

```
name [park]
    Speed = speed mph
    Height = height ft
    Length = length ft
    Status is status
```

- Only display the speed if its value is not the empty string. Only display the height if its value is not the empty string. Only display the length if its value is not the empty string.
- Here is an example with name, park, speed, height, length, and status:

```
10 Inversion Roller Coaster [Chimelong Paradise]
    Speed = 72 mph
    Height = 30 ft
    Length = 850 ft
    Status is operating
```

- Here is an example with name, park, height, and status (speed and length were empty strings):

```
Timberline Twister [Knott's Berry Farm]
    Height = 9 ft
    Status is operating
```

- Define the `displayFastestCoaster()` function.
  - Parameter: `coastersList` is a list of dictionaries where each dictionary represents a roller coaster
  - Return value: `None`
  - Call the `helper.getFastestCoaster()` function defined in the helper Python file to get a dictionary containing information about the fastest coaster.
  - Call the `displayCoaster()` function defined in this file to display it.
  - Here is an example:

```
Formula Rossa [Ferrari World]
    Speed = 240 mph
    Height = 52 ft
    Length = 2000 ft
    Status is operating
```

- Define the **displayAllParks()** function.
  - Parameter: `coastersList` is a list of dictionaries where each dictionary represents a roller coaster
  - Return value: `None`
  - Call the `helper.getParksList()` function defined in the helper Python file to get a list containing names (strings) of the unique parks.
  - Display the following message to the user:

```
Amusement parks in alphabetical order:
```

- Display each amusement park on a separate line followed by the number of unique amusement parks.
- Here is an excerpt showing the first three parks and the last three parks:

```
Amusement parks in alphabetical order:  
20th Century Fox World  
Abtenauer Bergbahnen  
Adlabs Imagica  
...  
Zygo park  
Ácqua Lokos  
Ölands Djur & Nöjespark  
There are 717 unique parks.
```

- Define the **displayCoastersInPark()** function.
  - Parameter: `coastersList` is a list of dictionaries where each dictionary represents a roller coaster
  - Return value: `None`
  - Have the user enter in text for a park. Allow the user to enter upper or lower case text, and allow the user to enter extra spaces before and after their input. Here is an example showing the user input in green text:

```
Enter a park:   berry farm
```

- Loop thru the list of coasters and check to see if the park text is in the name of the park. Make sure to handle the various cases (upper & lower) that the user's input and the data may contain. You also need to count the number of

coasters that meet the criteria. Display information about the coaster using the `displayCoaster()` function.

- Here is an example showing one of the coasters displayed when the park text is "berry farm":

```
Boomerang [Knott's Berry Farm]
Speed = 76 mph
Height = 36 ft
Length = 285 ft
Status is closed.definitely
```

- After the loop, print the number of coasters that were displayed.

```
Berry Farm has 11 coasters
```

- If there are no coasters that meet the criteria, then print a message. Here is an example when the park text is "balboa park":

```
No coasters in Balboa Park
```

- Define the `findCoasters()` function.

- Parameter: `coastersList` is a list of dictionaries where each dictionary represents a roller coaster
- Return value: `None`
- Have the user enter text for a search phrase. Allow the user to enter upper or lower case text, and allow the user to enter extra spaces before and after their input. Here is an example showing the user input in green text:

```
Enter a search phrase:  scooby
```

- Loop thru the list of coasters and check to see if the search text is in the name of the coaster. Make sure to handle the various cases (upper & lower) that the user's input and the data may contain. You also need to count the number of coasters that meet the criteria. Display information about the park using the `displayCoaster()` function.
- Here is an example showing the two coasters displayed when the search text is "scooby":



```
Scooby-Doo Spooky Coaster [Warner Bros. Movie World]
    Speed = 45 mph
    Height = 17 ft
    Length = 530 ft
    Status is operating
Scooby's Ghoster Coaster [Kings Island]
    Speed = 13 mph
    Height = 11 ft
    Status is closed.definitely
```

- After the loop, print the number of coasters that were displayed with the search term in single quotes. Here is an example:

```
Found 2 coasters that contain 'scooby'
```

- If there are no parks that meet the criteria, then print a message with the search term in single quotes. Here is an example when the search text is "Cookie":

```
No coasters contain 'cookie'
```

7. Create a Python file entitled **main\_*last*\_first.py** where *last* is your last/family name and *first* is your first name. Use lower case letters. Make sure to add the proper comment block at the top of the file.
  - Import the **helper** file in order for the main() function in this file to be able to call functions in the helper.py file that you have defined.
  - Import the **user\_interface** file in order for the main() function in this file to be able to call functions in the user\_interface.py file that you have defined.
  - Define the **getMenuDict()** function.
    - Parameter: None
    - Return value: a dictionary where the keys are letters for the user to input and the values are descriptions of the menu options.
    - The keys are the following letters: "A", "B", "C", "D", "E", "F", and "Q". The corresponding values (in order) are the following strings: "Number of coasters", "Number of operating coasters", "Fastest coaster", "Amusement parks", "Coasters in a park", "Find coasters", and "Quit".
    - Return the dictionary.

- Define the **main()** function.
  - Parameter: None
  - Return value: None
  - Display the following message to the user:

<b>Roller Coasters</b>
------------------------

- Call the `helper.createCoastersFromFile()` function to get the list of coasters. If your CSV file is named "roller\_coasters.csv", then you do not need to have an argument. Let the function use the default value for the filename. The `createCoastersFromFile()` function has a return value, so make sure to capture it in a variable. The return value is a list of dictionaries where each dictionary holds the information for one roller coaster.
  - Call the `getMenuDict()` function to get the dictionary for the menu.
  - Use a loop to display the menu, get the user's choice, and respond to the user's choice while the user does not enter "Q" or "q". To display the menu, use the `user_interface.displayMenu()` function. To get input from the user, use the `user_interface.getUserChoice()` function. Use branching to call the appropriate function in the `user_interface.py` file depending on the user's input.
- Don't forget to call the **main()** function.
8. Be sure to comment your code. This means that there should be comments at the top of the files you created as well as throughout your code. Put a comment block at the top of each Python file. Put a comment block before each function stating the parameters, return values, and what that function does. Points will be deducted for not having comments.
  9. Follow coding conventions. You should use lowerCamelCase or snake\_case for variable names. You are welcome to create any variables that you need.
  10. Test the program. Look at the Sample Output below. Projects that do not run are subject to 20% penalty. For the search, your output may be different if you are only getting the description to the first comma.

11. Prepare your submission:

- Find the **project\_last\_first** folder on your computer and compress it. This cannot be done within PyCharm. This folder should have the following files: helper.py, main\_last\_first.py, roller\_coasters.csv, and user\_interface.py. If you are completing the extra credit, which is described in the last 2 pages of this document, then you should also have a file named extra\_credit.py in the folder.
- On Windows, use **File Explorer** to select the folder. Right click and select the Send to -> Compressed (zipped) folder option. This will create a zip file.
- On Mac OS, use **Finder** to select the folder. Right click and select the Compress "*FolderName*" option. This will create a zip file.

12. Upload the zip file to your Blackboard section:

- On Blackboard, navigate to the Final Project item.
- Click on the specific item for the Final Project.
- Click on the **Browse Local Files** button and select the zip file.
- Click the **Submit** button.

## Grading

- This assignment is worth 100 points.
- Make sure that you the program runs. Points will be taken off if the graders have to edit the source code to test your program.
- Make sure to submit your assignment correctly as described above. Points will be taken off for improper submission.
- You may NOT import and use the CSV modules or other modules to process the CSV file. Use the techniques from the course materials.

Category	Item	Points
project_ <i>last_first</i>	roller_coasters.csv in directory	2
	helper.py	12
user_interface.py	getParksList()	5
	getFastestCoaster()	6
	displayMenu()	5
	getUserChoice()	5
	displayNumCoasters()	3
	displayNumOperatingCoasters()	5
	displayCoaster()	5
	displayFastestCoaster()	2
	displayAllParks()	4
	displayCoastersInPark()	8
	findCoasters()	8
	import helper & user_interface	2
main_ <i>last_first</i> .py	getMenuDict()	5
	main()	11
	<i>all files</i>	10
	style & comments	10
	submission	2
<b>Total</b>		<b>100</b>

## Sample Output

### Roller Coasters

```
A -> Number of coasters
B -> Number of operating coasters
C -> Fastest coaster
D -> Amusement parks
E -> Coasters in a park
F -> Find coasters
Q -> Quit
Choice: AAA
Choice:  a a
Choice:  a
The total number of coasters is 2796
```

```
A -> Number of coasters
B -> Number of operating coasters
C -> Fastest coaster
D -> Amusement parks
E -> Coasters in a park
F -> Find coasters
Q -> Quit
Choice: bee
Choice:  2
Choice: B
The total number of operating coasters is 2169
```

```
A -> Number of coasters
B -> Number of operating coasters
C -> Fastest coaster
D -> Amusement parks
E -> Coasters in a park
F -> Find coasters
Q -> Quit
Choice: c
Formula Rossa [Ferrari World]
    Speed = 240 mph
    Height = 52 ft
    Length = 2000 ft
    Status is operating
```

```
A -> Number of coasters
B -> Number of operating coasters
C -> Fastest coaster
```

D -> Amusement parks

E -> Coasters in a park

F -> Find coasters

Q -> Quit

Choice: **d**

Amusement parks in alphabetical order:

20th Century Fox World

Abtenauer Bergbahnen

Adlabs Imagica

Admiral Vrungel

Adventure City (Anaheim)

Adventure Island

...

ZDT's Amusement Park

Zhanggongshan Park

Zheshan Park

Zillertal Arena

Zoo Safari Stukenbrock

Zoo de Granby

Zoomarine

Zoosafari Fasanolandia

Zygo park

Ácqua Lokos

Ölands Djur & Nöjespark

There are 717 unique parks.

A -> Number of coasters

B -> Number of operating coasters

C -> Fastest coaster

D -> Amusement parks

E -> Coasters in a park

F -> Find coasters

Q -> Quit

Choice: **e**

Enter a park: **berry farm**

Boomerang [Knott's Berry Farm]

Speed = 76 mph

Height = 36 ft

Length = 285 ft

Status is closed.definitely

Coast Rider [Knott's Berry Farm]

Height = 16 ft

Length = 410 ft

Status is operating

GhostRider [Knott's Berry Farm]

Speed = 90 mph

Height = 36 ft  
Length = 1382 ft  
Status is operating  
Hangtime [Knott's Berry Farm]  
Speed = 92 mph  
Height = 46 ft  
Length = 670 ft  
Status is operating  
Jaguar! [Knott's Berry Farm]  
Speed = 56 mph  
Height = 20 ft  
Length = 793 ft  
Status is operating  
Montezooma's Revenge [Knott's Berry Farm]  
Speed = 89 mph  
Height = 45 ft  
Length = 244 ft  
Status is operating  
Pony Express [Knott's Berry Farm]  
Speed = 60 mph  
Height = 13 ft  
Length = 400 ft  
Status is operating  
Sierra Sidewinder [Knott's Berry Farm]  
Speed = 60 mph  
Height = 19 ft  
Length = 430 ft  
Status is operating  
Silver Bullet [Knott's Berry Farm]  
Speed = 89 mph  
Height = 45 ft  
Length = 953 ft  
Status is operating  
Timberline Twister [Knott's Berry Farm]  
Height = 9 ft  
Status is operating  
Xcelerator [Knott's Berry Farm]  
Speed = 132 mph  
Height = 62 ft  
Length = 671 ft  
Status is operating  
Berry Farm has 11 coasters

A -> Number of coasters  
B -> Number of operating coasters  
C -> Fastest coaster

D -> Amusement parks

E -> Coasters in a park

F -> Find coasters

Q -> Quit

Choice: E

Enter a park: happiest place

No coasters in Happiest Place

A -> Number of coasters

B -> Number of operating coasters

C -> Fastest coaster

D -> Amusement parks

E -> Coasters in a park

F -> Find coasters

Q -> Quit

Choice: disney

Choice: e

Enter a park: disney

Barnstormer [Disney's Magic Kingdom]

Speed = 40 mph

Height = 9 ft

Status is operating

Big Grizzly Mountain Runaway Mine Cars [Hong Kong Disneyland]

Speed = 56 mph

Length = 1100 ft

Status is operating

Big Thunder Mountain [Disneyland Park]

Speed = 60 mph

Height = 22 ft

Length = 1500 ft

Status is operating

...

TRON Legacy Coaster [Disney's Magic Kingdom]

Speed = 95 mph

Height = 24 ft

Length = 970 ft

Status is construction

Tron Lightcycle Power Run [Shanghai Disneyland]

Speed = 95 mph

Height = 24 ft

Length = 970 ft

Status is operating

Disney has 35 coasters

A -> Number of coasters

B -> Number of operating coasters



C -> Fastest coaster  
D -> Amusement parks  
E -> Coasters in a park  
F -> Find coasters  
Q -> Quit  
Choice: f  
Enter a search phrase: batman  
Batman [Furuvik]  
    Height = 10 ft  
    Status is relocated  
Batman [Malmö Folkets Park]  
    Height = 10 ft  
    Status is closed.definitely  
Batman [Tir Prince Family Funfair]  
    Status is operating  
Batman - The Ride [Six Flags New Orleans]  
    Speed = 80 mph  
    Height = 32 ft  
    Length = 823 ft  
    Status is closed.definitely  
...  
Batman: The Ride [Six Flags Discovery Kingdom]  
    Speed = 61 mph  
    Height = 37 ft  
    Length = 310 ft  
    Status is announced  
Batman: The Ride [Six Flags Fiesta Texas]  
    Speed = 61 mph  
    Height = 36 ft  
    Length = 310 ft  
    Status is operating  
Spinning Batman [Beijing Shijingshan Amusement Park]  
    Status is closed.definitely  
Found 17 coasters that contain 'batman'

A -> Number of coasters  
B -> Number of operating coasters  
C -> Fastest coaster  
D -> Amusement parks  
E -> Coasters in a park  
F -> Find coasters  
Q -> Quit  
Choice: f  
Enter a search phrase: cats  
No coasters contain 'cats'

A -> Number of coasters  
B -> Number of operating coasters  
C -> Fastest coaster  
D -> Amusement parks  
E -> Coasters in a park  
F -> Find coasters  
Q -> Quit  
Choice: quit  
Choice: exit  
Choice: q

## Extra Credit

Add additional features to your menu. Define and call functions to implement the features. For each option, you will need to update the getMenuDict() function and the main() function in the main\_last\_first.py file. You can implement one of the options or both of them. Add your new functions to a new file named extra\_credit.py. Make sure to include a comment block at the top of the file and comments throughout the file. Look at the sample output below.

Since this is for extra credit, you will not be given detailed instructions on how to implement these. You should be able to implement these on your own. Instructors and learning assistants are not required to help you with extra credit.

### Display the longest coaster – 4 points

- Define the getLongestCoaster() function to determine the longest coaster.
- Define the displayLongestCoaster() function to display the longest coaster which you will call in the main() function. Call your previously defined displayCoaster() function.

### Display the park with the most coasters – 6 points

- Define the getParkMostCoasters() function to determine the park with the most coasters. Call your previously defined getParksList() function.
- Define the displayParkMostCoasters() function to display the park and its coasters which you will call in the main() function. Call your previously defined displayCoaster() function.

## Sample Output for Extra Credit

### Roller Coasters

```
A -> Number of coasters
B -> Number of operating coasters
C -> Fastest coaster
D -> Amusement parks
E -> Coasters in a park
F -> Find coasters
G -> Longest coaster
H -> Park with most coasters
Q -> Quit
```

Choice: **g**

Alpine-Coaster [Golm]

Length = 2920 ft

Status is operating

A -> Number of coasters

B -> Number of operating coasters

C -> Fastest coaster

D -> Amusement parks

E -> Coasters in a park

F -> Find coasters

G -> Longest coaster

H -> Park with most coasters

Q -> Quit

Choice: **h**

1001 Pattes (Perc) [Foire]

Speed = 26 mph

Height = 3 ft

Status is operating

Achterbahn [Foire]

Status is operating

Ala delta [Foire]

Speed = 43 mph

Height = 16 ft

Length = 370 ft

Status is operating

Alligator Alley [Foire]

Status is operating

...

XenoX [Foire]

Height = 13 ft

Status is operating

Zading [Foire]

Status is operating

Foire has 119 coasters