

Project Instructions

To complete this project, follow the instructions below. If you get stuck, ask a question on Slack or in the Treehouse Community.

9 steps

- Create a new script

The very first step you will want to do after opening the workspace or unzipping the .zip file into your project folder is to create a new and blank Python script named `app.py` or `application.py`.

- Proper use of Dunder Main

Make sure the script doesn't execute when imported; Anything that is a calculation, callable function, or a block of logic that needs to run, ensure you put all of your logic and function calls inside of a dunder main block at the bottom of your file:

Dunder main statement looks like this:

```
if __name__ == "__main__":
```

HINT: Unit 1 project files/workspace had an example of this.

NOTE: This does not mean everything you write has to be contained within Dunder Main. You can still import and define functions outside of dunder main, you can still extract pieces of logic into those functions. The main calls to your program should be protected inside Dunder Main to prevent automatic execution if your script is imported.

- Import player data

Import from `constants.py` the players' data to be used within your program.

NOTE: Python has no concept of actual constants like other languages have. But it is a convention you may see in the real world. When you see ALL CAPS variable name you are meant to treat it as if it were a constant or a value that you cannot change/alter.

- Create a `clean_data` function

Write the logic to:

1) read the existing player data from the `PLAYERS` constants provided in `constants.py` 2) clean the player data without changing the original data (see note below) 3) save it to a new collection - build a new collection with what you have learned up to this point.

Data to be cleaned:

- Height: This should be saved as an integer
- Experience: This should be saved as a boolean value (True or False)

HINT: Think Lists with nested Dictionaries might be one way.

NOTE: Ensure you **do not directly modify the data in `PLAYERS` or `TEAMS` constants**. This data you should iterate and read from to build your own collection and would be ideal to clean the data as you loop over it building your new collection. If you are unsure of what this means, checkout this [instruction step](#).

- Create a `balance_teams` function

Now that the player data has been cleaned, balance the players across the three teams: *Panthers*, *Bandits* and *Warriors*. Make sure the teams have the same number of total players on them when your team balancing function has finished.

HINT: To find out how many players should be on each team, divide the length of players by the number of teams. Ex: `num_players_team = len(PLAYERS) / len(TEAMS)`

- Console readability matters

When the menu or stats display to the console, it should display in a nice readable format. Use extra spaces or line breaks (' \n ') to break up lines if needed. For example, '\nThis will start on a newline.'

- Displaying the stats

When displaying the selected teams' stats to the screen you will want to include:

- Team's name as a string
- Total players on that team as an integer
- The player names as strings separated by commas

NOTE: When displaying the player names it should not just display the List representation object. It should display them as if they are one large comma separated string so the user cannot see any hints at what data type players are held inside.

- An Example Run

Here is an example of what the running application might look like in the console (*the design of the menus and how things are displayed are totally up to you, though it should be readable and display the proper stats*)

```
BASKETBALL TEAM STATS TOOL
```

```
---- MENU----
```

```
Here are your choices:
```

- ```
1) Display Team Stats
2) Quit
```

```
Enter an option > 1
```

- ```
1) Panthers
2) Bandits
3) Warriors
```

```
Enter an option > 1
```

```
Team: Panthers Stats
```

```
-----
```

```
Total players: 6
```

```
Players on Team:
```

```
Karl Saygan, Chloe Alaska, Phillip Helm, Suzane Greenberg, Herschel Krustofski, Joe Smith
```

```
Press ENTER to continue...
```

- Before you submit for review

Make sure you can check off all of the items on the [Student Project Submission Checklist](#). The checklist is designed to help you make sure you've met the grading requirements and that your project is complete and ready to be submitted!

Extra Credit

To get an "exceeds" rating, complete all of the steps below:

5 steps

- Cleaning guardian field

When cleaning the data, clean the guardian field as well before adding it into your newly created collection, split up the guardian string into a List.

NOTE: There can be more than one guardian, indicated by the " and " between their names.

- Additional balancing to the team

Additionally, balance the teams so that each team has the same number of experienced vs. inexperienced players.

If this is done correctly each team stats should display the same number count for experienced total and inexperienced total as well as the same total number of players on the team.

- Include additional stats for a given displayed team:

- number of inexperienced players on that team
- number of experienced players on that team
- the average height of the team
- the guardians of all the players on that team *(as a comma-separated string)*

HINT: You can calculate the average height for a given team by keeping a running sum total of each players height on the team and dividing that total by the total number of players on that team.

- Quit Menu Option

The user should be re-prompted with the main menu until they decide to "Quit the program".

- An Example Exceeds Run

Here is an example of what the running application might look like in the console *(the design of the menus and how things are displayed are totally up to you, though it should be readable and display the proper stats)*

BASKETBALL TEAM STATS TOOL

---- MENU----

Here are your choices:

- 1) Display Team Stats
- 2) Quit

Enter an option > 1

- 1) Panthers
- 2) Bandits
- 3) Warriors

Enter an option > 1

Team: Panthers Stats

Total players: 6

Total experienced: 3

Total inexperienced: 3

Average height: 42.5

Players on Team:

Karl Saygan, Chloe Alaska, Phillip Helm, Suzane Greenberg, Herschel Krustofski, Joe Smith

Guardians:

Heather Bledsoe, David Alaska, Jamie Alaska, Thomas Helm, Eva Jones, Henrietta Dumas, Hyman Krustofski, Rachel Krustofski, Jim Smith, Jan Smith

Press ENTER to [continue...](#)

- NOTE: Getting an "Exceed Expectations" grade.
- See the rubric in the "How You'll Be Graded" tab above for details on what you need to receive an "Exceed Expectations" grade.
- Passing grades are final. If you try for the "Exceeds Expectations" grade, but miss an item and receive a "Meets Expectations" grade, you won't get a second chance. Exceptions can be made for items that have been misgraded in review.
- Always mention in the comments of your submission or any resubmission, what grade you are going for. Some students want their project to be rejected if they do not meet all Exceeds Expectations Requirements, others will try for all the "exceeds" requirement but do not mind if they pass with a Meets Expectations grade. Leaving a comment in your submission will help the reviewer understand which grade you are specifically going for