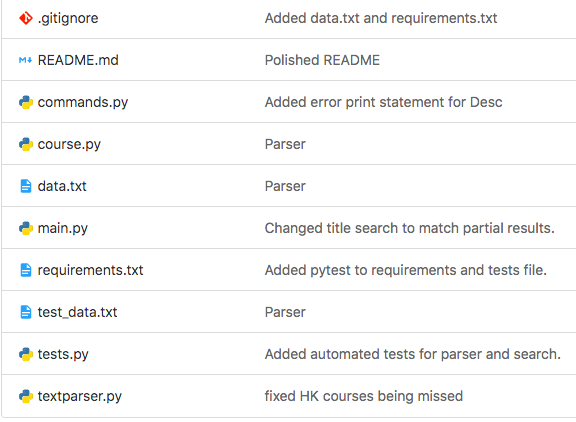
**Standards Document:**

**File System:**



README.md

This file is for information that any users of this project needs to know. How to set up what version of python is needed etc. Include any information that would be beneficial to the user here.

Commands.py

This is the location for the code that runs the commands issued by the user. All code related to manipulating or displaying the data from the parser goes here. The code written in here will be called from main.py. No commands should be written in any other files.

Course.py

This file contains the custom data structure for the course object. The representation of courses from the parser will use this format. The only use of this file is to contain the objects structure and to be imported by other files.

Data.txt

This file is an included and default example of the data. If the user does not provide their own file this one will be used instead.

Main.py

This file is used to contain the code that starts the program and takes in user input, it is also used to call the parser and the commands the user inputs. Only code dealing with user input should be included here.

Requirements.txt

This file is intended to be a list of requirements the user needs before using the project.

Test\_data.txt

This file is a separate data file used by the testing file in order to fill specific scenarios.

Tests.py

This file contains all of the unit tests for the project. Only code regarding testing should be included here.

textParser.py

This file contains the code to parse the data file, all functionality needed to parse the data into a list of the course.py object should be included here and nowhere else.

**Coding Standards:**

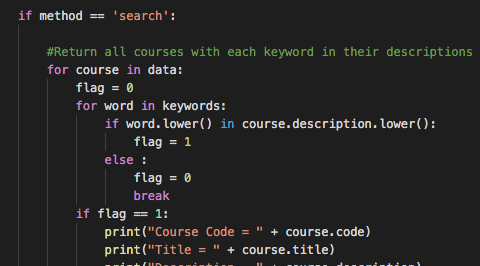
Python code should follow the python standards:

<https://www.python.org/dev/peps/pep-0008/>

Otherwise make sure indentation is correct and with correct syntax.

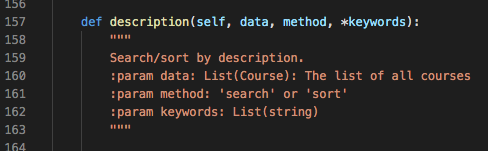
Start the scope of a block of code with a colon “:”

Ex.



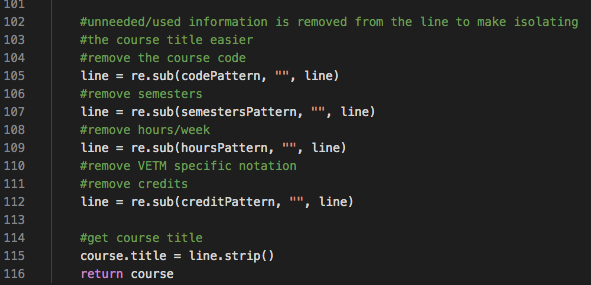
Each method will contain a detailed comment describing the function and parameters.

Ex.



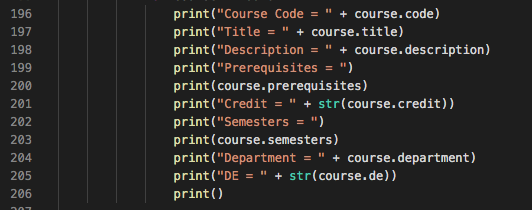
Comments used to describe logical decisions and complicated sections of code where needed.

Ex.



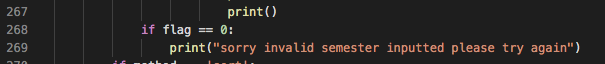
When outputting the courses info the following format is used.

Ex.



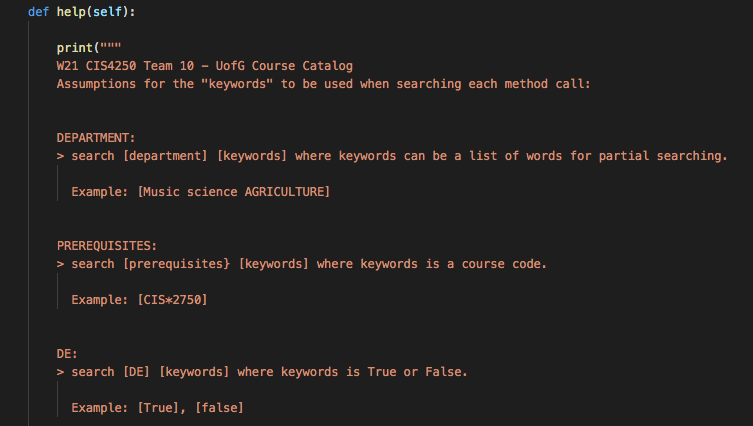
When encountering invalid input from the user include error text.

Ex.



Add useful information the user might want access to the help section of the commands.py

Ex.



**Development Process:**

Current Sprints end at 8am every Tuesday.

Next Sprint Planning every Tuesday morning at 9am.

Then continued at 3:30pm if needed.

During the sprint planning Team leader mediates the process.

Brainstorm issues and tasks that will be needed for the current sprint by Wednesday the following day. Assign issues and tasks after all the issues have been made. Make sure to evenly distribute the weight of the tasks to each member.

Source Control:

When starting on a task move your issue from “to do” on the sprint board to “doing”.

Also when starting on a new issue create a branch for that issue’s code changes.

Multiple issues should not be included on the same branch if possible.

Make frequent commits to the issue branch to prevent any loss of code.

When the issue is completed create a new merge request on the gitlab that requires one team member approval.

Let the team know you have a task done in the merge request section waiting for approval.

Fill out any documentation and testing needed for your particular task.

When another team member has approved your code you are then able to merge to master.

To avoid merging conflicts attempt to always have the most up to date version of master at all times.

Make detailed commits with meaningful titles, if you can include the issue numbers of the issue you are working on.

**Development Tools Used:**

Python 3.7+

Gitlab

Any additional tools should be added to this list.

**User Input/Output**

The user will need to input a specific command according to the structure in main.py presented to the user.

Ex.

W21 CIS4250 Team 10 - UofG Course Catalog

Use one the following commands to search/sort the data:

> search [parameter] [keywords]

> sort [parameter]

> help / quit

Where:

[parameter] is the course parameter you want to query.

'department', 'prerequisites', 'DE', 'description', 'credit',

'semester', 'title' or 'code'.

[keywords] is the keyword to use in the search.

Type 'help' for a detailed description of the allowed parameters and examples.

Type 'exit' to quit the program

The input and output used so far is all through text in the terminal.