INST 326 Final Project
Team: Programmers
INST Course Registration System
Mederick Grivel, Adam Isaacs, Noah Ramey, Sean Tran, Nadiyah Williams

Introduction To Our Project:
Our group decided to build a course registration system that would run in our computer's console. With this course registration, users can use the program both from a students perspective, and a school administrators prospective. Users can add new students to the school, enroll in new classes, view student statistics, and many other features that a university's school registration system might have.

Files In Our Repo:
There are 3 files in our repo (aside from the GitHub ReadMe and this documentation).

These files are as follows:
registration.py
registration.py is the main driver for our code, it contains all of the python script and is the file that you have to call in the terminal when you want to run the code.
courses.csv
courses is a csv file that contains sample data that the course registration will use to populate a dataframe that contains a section on each line. This file only contains 10 sections for simplicity's sake, but has the possibility to contain hundreds of sections (like a normal registration system may have).
students.txt
students.txt is a text file that contains student information. In our program, you can add new students one by one, but this is not effective at all for a class registration system. To combat this, our program allows users to upload a txt file with as many students as they want, and then we can enroll those students into courses and see their stats.

How To Use The Program:
To run the program, call the name of the script file, followed by the student's text file, and lastly the course information file. In the case of this repo, the syntax for calling and running our program would go as follows:

```
python3 registration.py students.txt courses.csv
```

Please note that student information must be formatted in a plain text file, and course information must be formatted as a csv file. Both files must also follow the correct line formatting so that our regex function can correctly pattern match each line.

Once you run our program, you will be asked which perspective you want to use the program from, you can select either a student perspective or a school administration perspective. Select

either perspective and you will be brought to a new menu, from here, depending on the perspective you selected, you will be able to access a variety of features.

Please note, the data for all students defaults to empty, to have the graphs display meaningful data, one must give grades to students which will then make impacts on their gpa and make the impacts of the methods more noticeable.

Attribution:

Noah Ramey:
parse_args()
get_grades()

Sean Tran:
student_stats()
__str__()

Nadiyah Williams:
class_ranking()
print_grades()

Adam Isaacs:
add()
get_gpa()

Mederick grivel:
regex_match()
addMultipleStudents()

*While more than 10 methods were written for this program, we have only attributed the methods that fulfill project requirements. Simple print methods have not been attributed.

Bibliography:
While our group did not have to use outside sources to complete any of our methods, many members of our group did refer back to pre-recorded lecture videos that Aric Bills, our INST 326 professor, published to his course page.