**ENSF 592 Spring 2023 - Assignment 2**

**📚 Learning Outcomes**

* Accept user input through varied menu options
* Validate user input through exception handling
* Process data according to specifications
* Design control flow logic using data comparisons
* Develop and implement user-defined classes
* Print formatted output according to given specifications

**💻 Program Specifications**

Computer vision and the related data processing allows cars to detect obstacles in their path. You are being asked to design a terminal-based application for determining a course of action depending on detected obstacles. Your application must meet the following design specifications:

* Your user interface should prompt the user to input the following information:
  + Select 1 to update the detected traffic light colour, 2 to update whether a pedestrian is detected, 3 to update whether a vehicle is detected, 0 to end the program
  + If menu option 1, 2 or 3 are detected, the user should then be prompted to specify the detected change
    - A traffic light can be "green", "yellow", or "red"
    - Pedestrian status can be "yes" or "no"
    - Vehicle status can be "yes" or "no"
  + A course of action message should be printed following the status change
    - Any scenario where a red light, a pedestrian or a vehicle are detected should display the message "STOP"
    - A green light with no pedestrian or vehicle detected should display the message "Proceed"
    - A yellow light with no pedestrian or vehicle detected should display the message "Caution"
  + After the action message, the current status of each monitored condition should be printed
* Your input interface design should follow the provided screenshot example.
* You must validate that the provided input is correct (both menu input and status input).
* If the menu option input does not meet the criteria, you must handle a ValueError exception by providing a message back to the user and allow them to re-enter their choice without terminating the program.
* All status input must match the given values exactly (e.g. "red" not "Red").
  + While you should check that the values are valid, you do not need to handle errors/exceptions for these values
* The initial default values are a green traffic light, no pedestrian and no vehicle.
* Your code should include and use the provided Sensor class and the provided user-defined functions. Provided code should remain unchanged unless otherwise specified. Details are provided in the template comments.
* You may not use any global variables. However, you may create your Sensor object in main.
* Your code must follow the conventions discussed so far in the course (names\_with\_underscores, ClassNames, four spaces for indentations, spaces between variables/operators, comments throughout, etc.)
* All user-defined functions must be properly commented above the function header.
* Your code will be run by the TAs as your end user.
* FAQs about the assignment will be answered on the D2L discussion boards. Please check the boards for any clarifications before submitting.
* The grading rubric will be posted to D2L.

**📝 Assignment Tasks**

* Make sure to watch video lessons 1 - 14 and review the corresponding Jupyter Notebooks and lab sessions.
* Clone this repository to your local computer.
* Open VSCode and start a new terminal. Make sure that your ensf592 environment is activated.
* input\_processing.py is provided as a starting point. Fill in the header with your own information.
* Remember to test your program execution via the terminal: python input\_processing.py
* Take a screenshot of your successful program run and upload it to your assignment repository.
* Commit your screenshot and code.
* Push your local git history to github
* Submit your repository HTTPS link to the Assignment 2 D2L dropbox.
* Tip: If you want to learn more about a specific aspect of a Python object, remember to take a look at the official documentation!

**About**

assignment2cardetectorp23-ESDiep created by GitHub Classroom

**Resources**

[Readme](https://github.com/Marasco-Teaching/assignment2cardetectorp23-ESDiep#readme)

**Stars**

[**0** stars](https://github.com/Marasco-Teaching/assignment2cardetectorp23-ESDiep/stargazers)

**Watchers**

[**1** watching](https://github.com/Marasco-Teaching/assignment2cardetectorp23-ESDiep/watchers)

**Forks**

[**0** forks](https://github.com/Marasco-Teaching/assignment2cardetectorp23-ESDiep/forks)

[**Releases**](https://github.com/Marasco-Teaching/assignment2cardetectorp23-ESDiep/releases)

No releases published

[Create a new release](https://github.com/Marasco-Teaching/assignment2cardetectorp23-ESDiep/releases/new)

[**Packages**](https://github.com/orgs/Marasco-Teaching/packages?repo_name=assignment2cardetectorp23-ESDiep)

No packages published  
[Publish your first package](https://github.com/Marasco-Teaching/assignment2cardetectorp23-ESDiep/packages)

**Languages**

* [Python100.0%](https://github.com/Marasco-Teaching/assignment2cardetectorp23-ESDiep/search?l=python)

**Suggested Workflows**

Based on your tech stack

1. Actions Importer

[Set up Actions Importer](https://github.com/github/gh-actions-importer)

Automatically convert CI/CD files to YAML for GitHub Actions.

Python Package using Anaconda

[Configure Python Package using Anaconda](https://github.com/Marasco-Teaching/assignment2cardetectorp23-ESDiep/new/main?filename=.github%2Fworkflows%2Fpython-package-conda.yml&workflow_template=ci%2Fpython-package-conda)

Create and test a Python package on multiple Python versions using Anaconda for package management.

Python application

[Configure Python application](https://github.com/Marasco-Teaching/assignment2cardetectorp23-ESDiep/new/main?filename=.github%2Fworkflows%2Fpython-app.yml&workflow_template=ci%2Fpython-app)

Create and test a Python application.

[More workflows](https://github.com/Marasco-Teaching/assignment2cardetectorp23-ESDiep/actions/new)

Top of Form

Dismiss suggestions

Bottom of Form

**Footer**

© 2023 GitHub, Inc.

**Footer navigation**

* [Terms](https://docs.github.com/site-policy/github-terms/github-terms-of-service)
* [Privacy](https://docs.github.com/site-policy/privacy-policies/github-privacy-statement)
* [Security](https://github.com/security)
* [Status](https://www.githubstatus.com/)
* [Docs](https://docs.github.com/)
* [Contact GitHub](https://support.github.com/?tags=dotcom-footer)
* [Pricing](https://github.com/pricing)
* [API](https://docs.github.com/)
* [Training](https://services.github.com/)
* [Blog](https://github.blog/)
* [About](https://github.com/about)