IS 430 – Foundations of Information Processing
Instructor: Kevin Trainor
Assignment: Final Project
Course Component: Final Project
Grading Rubric

Submission

Timeliness (10 available points)

Requirements
Must be submitted by date and time indicated in the weekly schedule.

Percent Credit	Description
100	On Time
50	Up to 3 days late
0	More than 3 and up to 7 days late
0	Not submitted or submitted too late

File Submitted (10 available points)

Requirements
Submit only 1 file.
File file submitted must be a .ZIP file that contains a PyCharm Project.
File and directory names must conform to the requirements stated in assignment instructions.
Data files must be located in the "data" subdirectory.
The "main_final_project_notebook.ipynb" file must be included.
The PyCharm project that you submit should contain all files necessary to test your code. This includes copies of any starter files.

Percent Credit	Description
100	Meets all expectations.
50	Meets nearly all expectations.
0	Does not meet expectations.
0	Not submitted or submitted too late.

Regular Portion of Final Project Assignment

Project Size (20 available content points)

Requirements
The effort demonstrated in this assignment should be equal to 2 coding assignments from the latter part of the course. When evaluating the effort demonstrated, we will consider the code that has been designed, written, and tested. We will also consider the learning effort required by the project. This includes effort expended on learning new Python libraries as well as effort expended on learning computing approaches to problem solving in the domain addressed by the project. Students should make sure to document these efforts well in their submission so that we can give them appropriate credit when grading is done for this section.

Percent Credit	Description
100	Meets all expectations.
90	Meets nearly all expectations.
75	Meets most expectations.
50	Meets some expectations.
25	Meets few expectations.
10	Meets nearly no expectations.
0	Meets no expectations.
0	Not submitted or submitted too late.

Other Measures of Completeness (25 available content points)

Requirements
The project submitted must include a copy of "main_final_project_notebook.ipynb" that has been completed to present appropriate details of the project.
Data for the project must be included and placed in the "data" subdirectory.
The project submitted must include an appropriate number of .py and .ipynb files that implement the project solution.

Percent Credit	Description
100	Meets all expectations.
90	Meets nearly all expectations.
75	Meets most expectations.
50	Meets some expectations.
25	Meets few expectations.
10	Meets nearly no expectations.
0	Meets no expectations.
0	Not submitted or submitted too late.

Technique (30 available content points)

Requirements
Include a Python Docstring that describes the intent of the program.
Place your highest-level code in a function named main.
Include a final line of code in the program that executes the main function.
Follow all PEP-8 Python coding style guidelines enforced by the PyCharm Editor. For example, place two blank lines between the code making up a function and the code surrounding that function.
Choose names for your variables that are properly descriptive.
Define CONSTANT_VALUES and use them in place of magic numbers.
Always use f-strings for string interpolation and number formatting.
When processing items from Python lists and tuples, unpack the values into variables with meaningful variable names to avoid using indexed expressions in your code.
Open all files in a WITH block to assure that they are closed before the conclusion of the program.
Remember that your program should behave reasonably when it is not given any input.
Model your solution after the code that I demonstrate in the tutorial videos.
Create a sub-directory named data within your PyCharm project to hold data files.
Remember to submit all data files with your PyCharm project – including the files that were provided as starter files to this assignment.
All functions that are not main() should have descriptive, action-oriented names.
All functions should be of reasonable size.
All functions should have high cohesion, and low coupling.
Remember to test your program thoroughly before submitting your work.
Your code must pass all relevant test cases. Make sure that it passes tests at the boundaries created by if, else, and elif conditions in your program (boundary value tests).
Use of the break statement is allowed but not encouraged.
Use of the continue statement is forbidden.
Regular expression patterns should be expressed as Python raw strings
Your finished code must be refactored to meet all good program design practices covered in this course.
When needed, custom Python classes should be created using Python Dataclasses using the approach demonstrated in our course.
Where it creates a practical advantage, manual unit tests should be replaced by automated unit tests created with Pytest.
Python programs that will be called from a Jupyter notebook, should not run when imported. They should only run when they have been explicitly called.
Python programs that are called from a Jupyter notebook should not prompt the user for input. Instead, configuration values should be set in a notebook code cell, then passed explicitly as parameters to a called Python function.
Python programs that will be called from a Jupyter notebook should be unit tested using PyCharm. This may be achieved using either manual unit testing techniques or by using Pytest automated unit testing.

Unit testing scenarios tested directly in PyCharm should not be repeated in the testing of the Jupyter notebook. Testing of the Jupyter notebook code should test the workflow that the notebook implements.
Jupyter notebooks should not include extensive Python code placed in the notebook code cells. When Python code in notebook code cells grows beyond the nature of configuration code, it should be placed into a .py library module file, imported into the notebook, then called from the notebook.
The Python code included in the notebook should be just enough to set configuration parameters, import Python functions that have been written and tested in PyCharm, and call those functions.

Percent Credit	Description
100	Meets all expectations.
90	Meets nearly all expectations.
75	Meets most expectations.
50	Meets some expectations.
25	Meets few expectations.
10	Meets nearly no expectations.
0	Meets no expectations.
0	Not submitted or submitted too late.

Challenge Portion of the Final Project Assignment)

Completeness and Technique (5 available content points)

Requirements
Content must be responsive to the directions given in the notebook prompt.
Writing must be persuasive.
Writing must be prose (rather than just bullet points) that follows professional standards for grammar, choice-of-words, and punctuation.

Percent Credit	Description
100	Meets all expectations.
90	Meets nearly all expectations.
75	Meets most expectations.
50	Meets some expectations.
25	Meets few expectations.
10	Meets nearly no expectations.
0	Meets no expectations.
0	Not submitted or submitted too late.

Total Available Points = 100