Individual Project

Using the knowledge that you have gained this semester; you will create a program in either Python or MATLAB. The intent of this project is for each student to demonstrate that they can apply their computing skills to address a topic or challenge that interests them. The topic for your project should connect with one of the following:

1. Your major or future major. What will you study and how could a program be useful?
2. Your future career. What might you need a computer program to do as a professional?
3. Your own interests. What might be interesting to have a computer program for? Choosing vacations? Assembling a sports team? Solving puzzles? Evaluating music? Other topics?

Each project must also meet the following specifications:

1. Brings in data either through reading a file or with user input (this can be as small as one value)
2. Outputs data by writing to the command window, producing plots or graphics, or writing to a file
3. Has a main program (script) that calls at least 3 user defined functions.
   a. At least one of the functions must be a different file (script) from the main program.
4. Includes at least three of the four following elements:
   a. One *for* loop
   b. One *while* loop
   c. One list, array, vector, or matrix
   d. One *if* decisional structure with at least one *else* or *elseif/elif*
5. Uses at least one case with nested structures which can be a combination of one or more loops imbedded in loops, one or more loops imbedded in conditional statements or conditional statements imbedded in one or more loops.
6. Employs at least two error checking algorithms for inputs, outputs, or intermediate data.
7. Meets or exceeds the minimum number of lines of active code (not counting comments)
   a. 100 lines for MATLAB
   b. 85 lines for Python

Exceptions to specifications: The intent of the specifications is to provide a standard level of complexity for each project. Since you all are choosing different ideas, the specifications provide that common standard. If you believe that your approach is sufficiently complex and challenging, you may request an exemption to one or more of the specifications. For example, if you are using complex logic and advanced functions that are well beyond the scope of ENGR 13300, your code may not have to be 100/85 lines. Only the faculty or the Graduate Teaching Assistants can grant this exemption and it will be done in writing. Requests for exemptions must be made prior to the day of demos. If you think you could qualify for the exemptions, ask a professor or GTA.

The Python or MATLAB files will be uploaded to Gradescope along with a report. Make sure that the files will run as uploaded. The report will consist of:

1. Title Page with your name, login, section number, and team number.
2. Introduction to the program and its theme that explains why you selected the topic and what it achieves.
3. Brief overview of the inputs and outputs. What will be required of the user and what will the receive and/or see?
4. Clear description of all user-defined functions.
5. User manual that includes screenshots, descriptions of what is needed to run the program along with sample inputs and outputs that help someone use the program.
6. An appendix with the code pasted into the document
7. Upload the Python or MATLAB files themselves into Gradescope in addition to the report.

Project Deadlines:

Proposal: Due BEFORE class on Thursday, November 11th

Demo Opportunity for extra-credit: November 30th, December 7th - 9th

Final Reports and Programs: Python/MATLAB files and report uploaded to Gradescope by Wednesday, December 15th by noon Eastern Time.

Proposal - 5 pts

5	Short description of the project idea

Project Grading Summary (see grading rubric below)

Program Code - 80 pts	
25	Code runs with no errors
55	Elements included
Report - 25 pts	
15	Description of program
7	Operating manual for code
3	Appendix includes the code
110	**Total**
	+5 bonus points for demo'ing yourself by Dec 9th +5 additional bonus points if ALL team members have demo'd codes by Dec 9th

Proposal Grading Rubric

Level Element	Fully Achieved (5 points)	Mostly Achieved (3 points)	Partially Achieved (1 point)	No Attempt (0 points)
Project Proposal	Project proposal is well-written and fully conveys how you plan to meet the goals of the project, makes connections to personal interest, major or future career, has few or no grammar and spelling mistakes.	Project proposal does not fully convey how you plan to meet the goals of the project or missing connection to personal interest, major or future career, or has several grammar and spelling mistakes.	Project proposal does not convey how you plan to meet the goals of the project in sufficient detail to provide feedback, missing connection to personal interest, major or future career.	Project proposal is not included

Program Code Grading Rubric

	Program Elements	
Score	Possible	
	15	Includes 3 from list (10pts for two, 5 for one)
		for loop
		while loop
		at least one list, array, vector, or matrix
		at least one *if* decisional structure with at least one *else* or *elseif/elif*
	6	Nested structures (loops in loops, if's in loops, loops in if's)
	15	Functions
		5 pts – at least one function outside of main program
		5 pts each for two functions within main program or outside main
	8	Input and output (4 pts input, 4 pts output)
	8	Error checking (4 pts for each error check)
	3	Comments (includes 1) academic integrity statement, 2) name and collaborators, 3) descriptions of key components of the program)
0	55	Total

Lines of Code

-15 points deducted if 85-100 lines in MATLAB or 70-85 lines in Python

-30 points deducted if less than 85 lines in MATLAB or 70 lines in Python

Running

___ 25 points code runs without errors

___ 20 points, code runs with error but has a work around to make it run or needs code changed during execution

___ 10 points, code runs partially but errors prevent from completing

___ 0 points, code will not run

Report Grading Rubric

Level **Criteria**	**Fully Achieved** **(5 points)**	**Mostly Achieved** **(3 points)**	**Partially Achieved** **(1 point)**	**No Attempt** **(0 points)**
Introduction to the program	Introduction is clear, includes motivation for the project and a brief description of the overall function of the project and is free of spelling and/or grammar errors	Introduction is clear, but does not include motivation for the project or a brief description of the overall function of the project or contains more than one spelling and/or grammar error	Introduction is not clear or contains more than 3 spelling and/or grammar errors	No program Introduction is included.
Description of the inputs and outputs	Clear overview describing the inputs and outputs and is free of spelling and/or grammar errors	Incomplete overview missing the inputs or outputs or contains more than one spelling and/or grammar error	Minimal description that does not provide enough substance to understand the inputs and outputs or contains more than 3 spelling and/or grammar errors	Overview is not included.
Description of user-defined functions	Clearly describes all user-defined functions	Clearly describes some user-defined functions but either does not include all functions or description is not clear for some functions, or contains more than one spelling and/or grammar error	Mentions that there are user defined functions but does not describe them or contains more than 3 spelling and/or grammar errors	Does not mention user defined functions
	(3 points)		**(1 point)**	**(0 points)**
Appendix (includes code)	Complete code is included in appendix		Partial code is included in appendix	Code is not included in appendix

	(7 pts)	**(4 pts)**	**(1 pt.)**	**(0 pts)**
User Manual	Clear and easy to understand user manual with concise descriptions and screen shots of the user interaction	Incomplete or hard to understand manual that either lacks clear instructions and/or does not include screen shots or contains more than 3 spelling and/or grammar errors	Started a user manual but is missing many required aspects	No user manual is included