

### 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 each student can apply the computing skills they have acquired to address a topic or challenge that applies to each student. The topic for your project should be one that connects with one of the following:

1. Your major or future major in engineering. 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 practicing engineer?
3. Your own interests. What might be interesting to have a computer program for? Choosing vacation destinations? Assembling a sports team? Solving logic puzzles? Evaluating music?

Each project should connect to your future major, your future career or personal interests and 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. 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.

3. Uses at least one for loop
4. Uses at least one while loop
5. Uses at least one case with nested loops

Note that the nested loops can have any combination of while and for loops and can count for one application of the loops (#3 and #4 but not both). The code must also use a loop somewhere else in the code.

6. Uses at least one vector or matrix
7. Uses at least one *if* decisional structure with at least one *else* or *elseif/elif*
8. At least 100 lines of code for MATLAB or 85 lines of code for Python (not counting comments).

**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 provides 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, as a professor or GTA.

The Python/MATLAB files along with a report will be uploaded to Gradescope. Make sure that the files will run as they are uploaded (e.g. have them all in the same folder). The report will consist of:

1. Title Page with section number, team number and team member names.
2. Introduction to the program and its theme that explains why you selected the topic and what it achieves.
3. Overall overview of program, function descriptions, description of inputs and outputs.
4. Clear description of all user-defined functions.
5. User manual that includes screenshots, descriptions, and sample inputs and outputs that help someone use the program.
6. Each student should upload the Python/MATLAB files directly into Gradescope. Include the code as an appendix in the final report.

**Project Deadlines:**

**Proposal:** Due Thursday, November 5<sup>th</sup>

**Demo Opportunity for extra-credit:** December 1<sup>st</sup> - 9<sup>th</sup>

**Final Reports and Programs:** Python/MATLAB files and report uploaded to Gradescope by Wednesday, December 9<sup>th</sup> by noon Eastern Time.

**Proposal Grading Summary** (see grading rubric below)

Points	
10	Short description of the project idea
<b>10</b>	<b>Total Proposal</b>

**Project Grading Summary** (see grading rubric below)

Program Code	
25	Code runs with no errors
55	Elements included
5	Comments in code
Report	
15	Description of program
10	Operating manual for code
5	Appendix includes the code
<b>115</b>	<b>Total</b>
	( 10 bonus points for each team member if ALL team members have demo'd codes are demo'd by Dec 9 <sup>th</sup> )

**Proposal Grading Rubric**

<b>Level</b> <b>Element</b>	<b>Fully Achieved</b> <b>(10 points)</b>	<b>Mostly Achieved</b> <b>(7 points)</b>	<b>Partially Achieved</b> <b>(3 point)</b>	<b>No Attempt</b> <b>(0 points)</b>
Project Description	Project description is well-written and clear, has few or no grammar and spelling mistakes, and conveys how you plan to meet the goals of the project	Project description is not clear, or has many grammar and spelling mistakes, or does not fully convey how you plan to meet the goals of the project	Project description is very incomplete and does not contain enough information to provide feedback	Project description is not included

**Program Code Grading Rubric****Required Elements (5 Points each)**

- Brings in data either through reading a file or with user input (this can be as small as one value)
- User-defined function #1
- User-defined function #2
- User-defined function #3
- Uses at least one for loop
- Uses at least one while loop
- Uses at least one case with imbedded loops  
Note: imbedded loops can have any combination of while and for loops and can count for one application of the loops (#3 and #4 but not both). The code must use a loop somewhere else in the code.
- Uses at least one vector or matrix
- Uses at least one if structure with at least one else or elseif

**Subtotal** \_\_\_\_\_ (45 if all present)

**100/85 Lines of code\***

*(\* Or Pre-approval from GTA/Professor for sufficiently complex project).*

- 10 points for  $\geq 100$  MATLAB or  $\geq 85$  Python
- 5 points for  $70 < \text{lines of code} < 100$  for MATLAB or  $60 < \text{lines of code} < 85$  for Python
- 0 points for less than 70 for MATLAB or less than 60 for Python

**Comments**

- 5 points, well commented, uses the templates and add some comment lines within the code
- 3 points, uses the templates with comments but nothing else
- 1 point, comments present but incomplete
- 0 points, no comments

**Total (Elements + Lines of code + Comments)** \_\_\_\_\_/40

**Running**

- 25 points Codes runs without errors
- 20 points, code runs with error, but can get code to complete easily working around error
- 10 points, code runs partially but errors prevent from completing
- 0 points, code will not run

**Report Grading Rubric**

<b>Level Criteria</b>	<b>Fully Achieved (5 points)</b>	<b>Mostly Achieved (3 points)</b>	<b>Partially Achieved (1 point)</b>	<b>No Attempt (0 points)</b>
<b>Introduction to the program</b>	Introduction is clear and free of spelling and/or grammar errors	Introduction is clear but 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.
<b>Overview of program</b>	Clear overview of overall program, describes its function, including inputs and outputs	Incomplete overview of overall program, or does not describe its function, including inputs and outputs	Minimal description that not does provide enough substance to gain an overall understanding of program	Overview is not included.
<b>Description of user-defined functions</b>	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	Mentions that there are user defined functions but does not describe them	Does not mention user defined functions
<b>Appendix (includes code)</b>	Complete code is included in appendix	Partial code is included in appendix		Code is not included in appendix

	<b>(10 pts)</b>	<b>(5 pts)</b>	<b>(1 pt)</b>	<b>(0 pts)</b>
<b>User Manual</b>	Clear and easy to understand user manual with screen shots of the user interaction	Incomplete or hard to understand manual that either lacks clear instructions and/or does not include screen shots	Started a user manual but is missing many required aspects	No user manual is included