

ENGR 102 – Spring 2022 (Cruzado)
Project Specification

Deliverables:

There are several deliverables for this combination team and individual project. Please submit the following file to the Mimir and eCampus Submission Box labeled “Project Plan” by **28 March 2022 at 11:59 pm**:

- `Project_Plan_TeamXXX-X.pdf` where XXX-X refers to your team number

Please submit your completed project (all `~.py` file(s) and PDF documents) to the Mimir and eCampus Submission Boxes labeled “Final Project – Team” and “Final Project – Individual” by **03 May 2022 at 11:59 pm**.

It has **300 points** in total – 200 for the team portion and 100 for the individual portion.

At the end of the course (**02 May 2022**) there will be an online presentation of the project with **70 extra points**.

Project Plan: Planning your project – to be done as a team (50 points)

Engineers today are expected to maintain a combination of technical problem-solving capabilities, content literacy, and societal skills of communication, creativity, and collaboration¹. This project is meant to improve your creativity skills while also working on your technical problem-solving capabilities within the framework of a self-defined project. Your team is first tasked with coming up with a project idea – as a team you will create a program that does something of your choice. Please see the file **Project_Ideas.pdf** for suggestions.

Regardless of the specific application of your program, it must meet a set of minimum requirements, defined below. At a minimum, your project program is required to do the following:

- Display the purpose and a set of instructions of how to use your program
- Display a set of menu options for the various things your program can do
- Use at least one if-elif-else statement
- Use at least one loop
- Use multiple functions with docstrings
- Use at least one try-except block for user input validation
- Use file input/output for something
- Does something (solves a problem, provides entertainment, ...)
- Incorporates at least one thing beyond what is covered in the lectures (learn something new)
- Be creative and have fun!

You are encouraged to add extra features to your program, but please submit a working file for grading.

As a team, decide on a project idea. Write a description of your idea (in complete sentences, minimum 500 words). This should include the motivation of why your team chose the idea, any background information to help understand the purpose of your idea, and details about what you want your program to do.

¹X. Du, M. Thrane, M. Lehmann, P. Christensen. Problem-oriented and project-based learning (popbl) as innovative learning strategy for sustainable development in engineering education. *European Journal of Engineering Education*, 33(3):283–295, 2008.

Project Specification

After your team decides on a project and writes a description, start thinking about how your program will work. Using the top-down approach, create a hierarchy for the design of your program. You can draw it by hand, or use software of your choice. Make sure the leaves of your hierarchy can each be implemented in just a few lines of code. This won't be your final hierarchy; it's just a draft. As you begin to code you will likely change your mind about how you want to implement the various aspects of your program. You may not know exactly how to do something right now, but you will learn along the way.

Create a document named **Project_Plan_TeamXXX-X.pdf** (where XXX-X is your team number) that contains your team's description and hierarchy.

Final Project (Team): Writing your code – to be done as a team (150 points)

After your project idea is approved, divide up the workload among your team so that each team member can write one portion of the program on their own. If you planned your program well during the planning stage, each portion can be written and tested individually. Remember to write test cases *before* you write code and use the pyramid style of code development.

After each team member has written their part, combine the code into one or more files as needed. As a team, work together to complete the final bits of code and testing needed to debug your program. When you are done, have each team member independently verify that the program works as desired.

In addition to submitting a **working ~.py file**, your team must also develop a pdf document of how to use your program – a **user manual**. This document must describe how to run the program, include examples of valid inputs and expected outputs, and a description of how your code works. Including screenshots or flowcharts is a good idea.

Final Project (Individual): Reflecting on the design process – to be done individually (100 points)

Create a document named **Final_Project_yourname.pdf** and answer the following questions. Please fill in your first initial and last name for the text yourname. For example, Final_Project_Acruzado.pdf. **Please write in complete sentences.**

1. Did you like your project topic? Why (not)?
2. Did your team stick with your initial top-down approach to program development, or did you switch to a different design approach at some point? Which approach did you prefer and why?
3. Did your final design closely match the original hierarchy your team developed? In what ways is it the same? In what ways is it different?
4. Briefly describe the thing(s) you learned on your own, beyond what was covered in the lectures. How did you incorporate them into your program?
5. If you had more time to work on your program, what additional features would you want to add? Is there anything about your submitted program you would change?
6. What was the most difficult part of this assignment? Please explain.
7. Estimate the portion of the assignment completed by each member of your team (yourself included). Please explain any significant workload imbalances and give a brief summary of who did what. For this question it's ok to make a bulleted list instead of writing complete sentences.

Example:

- a. Avery: 25% - completed hierarchy and pdf document, programmed instructions functions
- b. Bailey: 35% - really enjoyed working on the project and took over, created list of functions and variables used, created test cases, coded almost half of it, put code together and debugged it

Project Specification

- c. Cameron: 15% - sick and busy studying for other classes, programmed one function
- d. Dylan: 25% - programmed input validation and menu options, helped with debugging

Project presentation – to be done as a team (70 points). All members of the team must present to get the extra credit.

Please submit your completed project presentation to the Mimir and Ecampus Submission Box labeled “Project Presentation” as **Project_presentation_Teamx.ppt** where x is your team number. The presentation should include:

4-5 slides:

1. Your program title
2. Short description
3. Motivation for choosing the specific game
4. Status of the program
5. New programming feature that you have included (not taught in class)
6. Difficulties you may have developing your program

In class you can show me a demo of your program