CS 108 Final Project (v1.~~0~~1)

Fall 2025

Rocky K. C. Chang
~~October 24~~November 9,
2025

The Objectives

The overall objective is to demonstrate that you are able to apply what you have learned in this course through this self-initiated project. In particular,

1. Acquire skills and experience of working on a Python project independently.
2. Apply sound methodologies and practices to solve problems.
3. Elevate your Python programming skills by engaging in a larger codebase.

Project team

Each project team consists of one or two members who can come from different sections. Sign up here. Only one submission is needed from each group.

How and where to start?

Choose your own topic which could be

1. Find something from your major, if it is not CS, that can be developed into a computer program.
2. Look through some previous projects in Moodle.
3. Inspired by the lab/homework you did in the class. Possible extensions? (e.g., a more complicated encryption algorithm or game)
4. Useful websites:
    - Book on guizero: https://magazine.raspberrypi.com/books/create-guis
    - Some project ideas: https://inventwithpython.com/blog/new-book-big-book-python.html (some projects there are too simple though.)
5. You can come to discuss it with me or the tutors in the study sessions during the formulation of your proposal.
6. If you want to pick up more Python stuff for your project, I recommend Real Python.

The project should not be too easy or too difficult and can be completed by your team in 6 weeks.

- Start from small and then grow it gradually.
- Better to finish a not-so-difficult project than not finishing at all.

Previous projects

Utilities/tools, e.g.,

- Dynamic Task Management System: Enhancing User Productivity
- Electronic Component Analyzer

Music/art related, e.g.,

- Audio Signal-Processing Module with Simple Graphical Interface
- Creating a Simple Drawing Application through Guizero

Recommendation systems, e.g.,

- Analyzing Users Restaurant Review through Word Counts and Sentiment Analysis
- Recommending Christian movies based on the description of the contents of the Bible

Education, e.g.,

- Meet the Oaks: An Identification Tool for Oak Trees on Calvin's Campus
- Teaching Children How to Punctuate Through a Simple Text Analysis
- NutriCatch – Learn Healthy Eating habits through a game

Games, e.g.,

- Rock-Paper-Scissors Game Against an Intelligent RPS Player
- A Hangman Game, Including Different Difficulties and Game Statistics
- King Taco as a Card Game Web App

Algorithms and simulation, e.g.,

- Maze Solving using Breadth-First Search in Python
- Cribbage In Python: The Best Hand Algorithm
- Maze Generation Using the Wave Function Collapse Algorithm

Requirements and expected amount of work

- Use Python 3.*.
- Use `turtle` for simple graphics.
- Use `guizero` module for GUI.
  - Strong justifications required for using other more advanced modules, such as `tkinter`, `pygame`, etc
- Sufficient problem-solving effort
  - Identify the data and use appropriate Python data types.
  - Develop algorithms for solving problems.
- Sufficient coding effort
  - You could use existing modules to speed up your development process.
  - Acknowledge the modules written by others (except for the modules used in class).
  - Acknowledge any help you receive from others.
- ~35-60 hours spanned over around 6 weeks

The milestones and assessments

The project is made up of 8 milestones which could help you make steady progress in completing your project well. Due to the tight schedule, there will be no extension. The policy for late submission penalty applies here.

0. Form a team of at most 2 members: 0% (due at 23:59 on Oct 27)
1. Project proposal: 10% (due at 23:59 on Nov 3)
2. Progress report: 10% (due at 23:59 on Nov 10)
3. Code submission 1: 10% (due at 23:59 on Nov 17)
4. Code submission 2: 10% (due at 23:59 on Nov 24)
5. Code submission 3: 10% (due at 23:59 on Dec 5~~1~~) – no late submission accepted
6. Report and code submission: 45% (due at 23:59 on Dec 11~~8~~) – no late submission accepted
7. Showcase in the lab: 5% (same as the exam period)

1. The project proposal

The proposal is of at most 2 pages with reasonable margins and font size to be submitted to Moodle by the deadline. Assessments:

1. Motivation: 5%
2. A specific and informative title: 20%
3. Goals for the project: 25%
4. Specifications for the final app: 25%
5. An execution plan: 20%

6. Python modules required: 5%

For each item, 0-4 points will be given:
4: Exceptionally well thought out and well written
3: Good quality in the write-up
2: Acceptable quality in the write-up
1: Very vague and unclear write-up
0: No submission

2. Project progress report

The progress report is of at most 5 pages with reasonable margins and font size to be submitted to Moodle by the deadline. Assessments:

1. An updated title: 10%
2. Updated goals: 20%
3. Updated specifications for the final app: 20%
4. Design of the solution: 50%
   - (15%) Describe the data in your project. What kinds of Python data types will you use?
   - (20%) What are the main challenges in your project? What is your plan to tackle them?
   - (15%) A (partial) list of functions (classes and methods for OO) in your program design
5. Help from other people (put it at the end of your report)
   a. Acknowledge in details (including the links if applicable) the help you received from other people.
   b. The help for those ideas/design used in your project (e.g., inspired by so and so's work)
   c. Failure to report will result in mark deduction which depends on the extent of the unreported parts of your work

For item 1-3, same as the project proposal
For item 4:
4: Exceptional quality in the design
3: Good quality in the design
2: Acceptable quality in the design
1: Very vague and unclear design
0: No submission

Code submission (for code submissions 1-2 and the final code submission)

If you have more than one `.py` file. Name the main file by `main.py`. Whether you submit one or more files (.py + files with other file extensions), compress all of them into a zip file and name it by `lastname.zip` for single person and `lastname1_lastname2.zip` for 2-member group. Pay attention to the following points:

1. Your program will be tested only on Thonny. Therefore, make sure it works on Thonny.
2. If your program uses files (read/write), do not specify paths to your files.
3. You need to submit all other files, if any, with your .py files.

3. Code submission 1

Assessment:

1. A main program written? (30%)
2. The signatures of all functions defined (including the docstrings)? (30%)
3. Amount of code written? (30%)

4. Running the program without errors? (10%)
    - Use the keyword `pass` for unfinished function/method implementation.
5. Help from other people (put it at the beginning of your main program in docstring)
    a. Acknowledge in details (including the links if applicable) the help you received from other people. Recall that using generative AI is not allowed in this course.
    b. The help for those techniques/design used in your project that are not covered in our lectures, POGILs, lab and homework
    c. Failure to report will result in mark deduction which depends on the extent of the unreported parts of your work.

For items 1,2,4:
4: Yes
0: No

For item 3:
4: Most of the functions/classes implemented
3: A good number of functions/classes implemented
2: Some functions/class implemented
1: Only a few lines of code written
0: No code written

4-5. Code submission 2-3

Code submission 2 will be evaluated based on the improvement over code submission 1 (e.g., additional code, bug fixing, better code documentation). The improvement must be listed at the beginning of the main program in a docstring. Code submission 3 will be evaluated similarly.

4: Exceptional improvement
3: Good improvement
2: Some improvement
1: A little improvement
0: Same as before

6. Final report and code (no late submission accepted)

The final report:

The report is of at most 8 pages (11 pts Times New Roman or similar), single space and 1 inch for all four margins. The report must contain the following sections. You must submit it in pdf.

1. An informative and concise title and your name (5%)
    - All the keywords should appear in the title.
2. Novelty (5%)
    - What's so special about this project (as compared with other similar projects)?
3. Goals (10%)
    - What problem are you solving?
    - The overall goals and more specific subgoals
4. Data and Python data types (10%)
    - What kinds of data are relevant to solving this problem?
    - What Python data types do you use to implement them?
5. Algorithm design (10%)
    - Describe the algorithms you use to solve the core problems.
6. Program design (10%)
    - A table of functions/classes you will have designed and their relationship

7. Failure to comply with the report format requirements will result in 5% penalty.

The final code:

Your code will be evaluated based on

1. Correctness of the program (i.e., correct results, free of crash, graceful handling of various exceptions, etc): 10%
2. Use of what you have learned in class: 10%
3. Code readability and documentation: 10%
4. What you have learned by yourself for this project (e.g., techniques/design used in your project that are not covered in our lectures, POGILs, lab and homework or learning the Python topics that have not been covered in the class). You must include them at the beginning of the main program file in a docstring: 5%
5. Coding effort: 10%
6. Easy to use and navigate (i.e., state the purposes of the app, clear instructions of using the app, etc): 5%
6. Help from other people (put it at the end of your report)
    a. If you do not get any help, you still need to include this section and state so.
    b. You do not need to acknowledge Rocky's help (because he knows it already) and the imported modules.
    c. Acknowledge in details (including the links if applicable) the specific help you receive from other people (including video, articles, and discussion forums)
    d. You are still expected to originate most of the code.
    e. Failure to report will result in mark deduction (at least 10%) which depends on the extent of the unreported parts of your work
    f. If time permits, you will be interviewed for authenticating your work.

Each item in the final report and code assessment will be rated in a 4-point scale:

4: Exceptional
3: Very good
2: Good enough
1: Could do much better
0: Unacceptable quality

7. The showcase

1. Get everyone into the rows, each has 4 or 5 projects.
2. Spend around 20 minutes to have each row show-off to each other with the goal of coming up "the best project."
3. Each best project from a row will be demonstrated to the class.
4. Vote for the best project.
5. Eat donuts (or other suggestions?)
6. Students attending the showcase will normally be awarded full marks.
7. No additional marks will be awarded to the best projects.

Academic integrity

1. Feel free to discuss ideas with Rocky or with your classmates/tutors/friends/relatives/…. Please acknowledge and detail the kinds of their help in your report.
2. Write your own code.

3. Note that ChatGPT, Claude and other similar generative AI systems are NOT allowed in this introductory course for the reasons explained in the class.