

Assignment 2 – Final Project

Overview

Assignment 2 is an opportunity for you to write an **original Python program**, which showcases your Python programming skills and creative flair. The main objective is to apply the knowledge and skills you have gained during this module to create a program which either solves an interesting problem or provides engaging functionality. An example problem could be that you have a set of data files which need to be analysed. An example functionality could be a fun game that the user can play.

You are free to choose any problem or functionality you would like and are highly encouraged to pick something which interests/excites you. If you are not sure your project idea is feasible, please ask your TA for feedback. If you would like some inspiration there is a list of standard project ideas (see below) and a good execution of one of these standard ideas would get you a decent, but unspectacular, mark. To reach top marks you will need to either significantly expand upon one of the standard ideas, or implement a good execution of a novel idea.

This assignment is worth 60% of your overall mark for the module EMAT10007.

Deliverables

The deliverables for this assignment are:

1. A working Python program in the form of a .py file. Alternatively, a collection of files which are collected together as a .zip folder.
 - Programs will be tested in **Python 3**.
 - Programs must successfully run on the University computers. Please check this before submitting.
2. A short report in the form of a .pdf file which is 1-2 pages in length. It should do the following:
 - Describe the goal and motivation behind your project.
 - Cite your sources. Mention any external code used, making the URLs clear.
 - Describe how to use the program.
 - What design choices did you make? Why did you choose certain solutions over others? For example, why a dictionary over a list? Or why did you use a certain module? [**Very important**]

- Any challenges you encountered and how you over came them. Please include at least one example. [**Very important**]
- Anything else you would like to highlight, and that your marker should know in order to give you a correct and fair mark.

Submission Info

The assignment deadline is **3pm on Friday 13th December**.

You must upload your assignment to **Blackboard**. Multiple submissions are possible - we will only mark the latest submission before the deadline.

If you are unable to submit by the deadline, you will need to request an extension **before the deadline has passed**. All extension requests are handled by the School Office. You will need to complete an 'Extension Request Form' which is available in the 'Engineering Student Forms' folder on the Engineering Faculty Student Intranet. Please note that technical problems such as computer malfunctions or loss of data are **not valid reasons** to get an extension. There are many computers available on campus and it is your duty to back up your work regularly.

For this assignment it is required that you submit **individual work**. You may discuss the create scope of the assignments with each other, but you must work individually on the program and report. **Note that we will be cross-checking projects.**

Assignment Criteria

This assignment is to assess your ability to apply the knowledge and skills you have gained during this module. The following outlines some expected criteria of a good submission.

Good programs should:

- Apply learned concepts in a useful and interesting way (i.e. use variables, loops, conditions, functions, built-in data structures, modules, user interactions, etc.)
- Include helpful comments (e.g. describe functions, explain hard to grasp logic, etc.)
- Use sensible and informative variable names (e.g. use of CamelCase, consistent naming convention, self-explanatory names, etc.)
- Have clear structure making the code easy to read and understand.
- Use loops, functions and classes where appropriate to simplify the structure of the code, making it tauter and avoiding excessive duplicate code.
- Be robust to different inputs and provide appropriate feedback to the user.

For higher marks, programs should achieve some of the following:

- Have an original idea (i.e. not one of the standard ideas provided, see below).
- Employ extra functionality from a number of different built-in modules.
- Use external modules like Tkinter, PyGame, Matplotlib, NumPy, etc. (We will cover these in the remaining lectures and labs.)
- Use data structures beyond the standard built-in data structures.
- Use a simple AI in the case of a game.
- Save and read relevant data to/from files (e.g. high scores, status of the program, last data points, etc.)
- Use a Graphical User Interface (GUI), e.g. with Tkinter.

Exceptional submissions may include:

- Implementation of an outstanding and original idea.
- Use of a complex multiple file structure.
- Creation and use of your own classes.
- Use of modules not explicitly shown in the lectures (e.g. Pandas, PySpice, BeautifulSoup, etc.)
- Use of a more sophisticated Graphical User Interface (GUI).

TIP: Start with a simple program (i.e. basic functionality), get this working and then add more features and complexity. Make sure to start working on the project early - don't leave it until the last minute - and to backup your data in multiple places.

Lab Time

We will set aside time to work on your projects and ask questions in the lab sessions and drop-ins. Note that it is expected for a top-end assignment to take more time than allocated during lab sessions. Please make sure you are being effective with the time in the labs and ask questions while the TAs are available.

Plagiarism

It is very easy for us to run Python code through plagiarism checking software that analyses for similarities between all the submitted assignments and between those assignments and other code available on the internet. The University penalties for plagiarism are very unpleasant - so please do not be tempted to cheat.

We will check every submission for plagiarism!

It is fine to research ideas, or develop an idea that you find on the internet, but you should avoid copying large chunks of code. If you are basing your project on something that you have found on the internet, you should describe in your report exactly where you found the idea (i.e. provide a link) and how your contribution is different.

We can only mark code which has been written by you.

Any use of code found online should be highlighted and referenced in a comment and your report. If you are unsure about anything, please check with your TA before submitting.

Project Ideas

A bare minimum project idea would be a simple text interaction game. Note that as these ideas really are the bare minimum expected work, they will need to be executed flawlessly to pass. Examples include:

- Various card games such as Poker, Solitaire, etc.
- Maze solving (read different text-based mazes from a file, display it and solve it for the shortest route).
- Two-player Chess, Connect Four, Drafts or other board games. (N.B. Programming a ‘computer player’ is very difficult.)
- Scrabble - asking the player to find the best scoring words from a randomly generated rack. (N.B. You can find databases of words online).

More interesting project ideas (and therefore capable of achieving higher works) would be:

- A physical simulator (using odeint for integration).
- A complex calculator capable of computing statistics with a GUI. (N.B. There are lots of examples of this online, so make sure that your code is original.)
- Extracting, analysing and plotting data from data files (with user interaction).
- A drawing application.
- A game implemented with PyGame.

If you are not sure if your idea is valid, please check with us. We can give feedback on how difficult it might be and point you to useful Python modules. **It is mandatory that everyone checks their project idea with their TA during the lab session on Friday 29th November.**

Note: You should be ok using modules which are built-in to Python 3 (the standard library) and any presented in lectures e.g. PyGame, Tkinter, NumPy and matplotlib. For some more interesting projects you might want to experiment with downloading third part modules. You can do this within PyCharm - see the following for more information: <https://www.jetbrains.com/help/pycharm/installing-uninstalling-and-upgrading-packages.html> If you add any non-standard packages, double check it with your TA and mention this in the report.