MT2505 Computing Project: Guidance on what to submit

MRQ[*]

Semester 2, 2019–20

This document explains what you should submit when you have completed the two parts of the Computing Project for MT2505.

You should submit one file containing your solutions to both parts of the project, via MMS no later than 7pm on Friday 24th April (the end of Week 11). Your submitted file should be a `.py` file that contains a combination of executable Python code and comments.

Important points

Some important points to keep in mind:

- Beware of the difference between `print` and `return`!
- Submit only one, plain text file, with extension `.py`.
- The submitted file must be valid Python 3 code and not Python 2.
- Clearly identify the solution to each question with a comment, for example, `# Question 1`.
- You should submit code that answers each of the questions labelled **Question ?** on the project sheets.

  Do not submit solutions to the other questions that appear (particularly those labelled **Check and Debug Your Code**). These additional questions are to aid you in your project, but are not assessed.
- The first thing the markers of the project will do is execute the submitted file. If it cannot be executed, for whatever reason, then marks will be deducted. The file should *not contain any syntax errors* and *no errors* should occur when it is executed.

  The submitted file should only take one or two minutes to run, and you will be penalised if it does not finish in *under 20 minutes.* You will not be penalised if it takes anywhere under 20 minutes to execute.
- *Do not use* `print` *statements* anywhere in the submitted file. Marks will be deducted if the submitted file contains any `print` statements.

  If you use `print` statements for debugging, then *remove them* before submitting your final debugged code.
- The exact and complete Python code required to solve each problem should be contained in the submitted file.

[*] Based on guidance produced for previous years by Prof. J. D. Mitchell

- The expected output should be indicated with a *comment*. You should not include any of the expected output in the file outside of comments (if for no other reason, because most likely this might cause the code not to execute).

  After the code that performs the calculation, insert a comment of the form

  ```
  # Returns ???
  ```

  (replacing ??? by your actual answer).

- The submitted file should be *self-contained*, which means that everything that is required to run the file is included in the file. The only files that should be loaded (using `import`) should be those provided along with the project and (when necessary) standard Python modules.

  You must `import` some of the files provided with the project (for example, to load the command to produce the binary operations on a set of size 3) if needed by your code. If your code depends upon such a file and you do not `import` it, then your code will probably not run correctly and you will lose marks.

- Do not confuse the strings `"True"` and `"False"` with the boolean values `True` and `False`.

- You should ensure that when your submitted file is run that it performs all of the calculations required to answer the questions in the project.

It was observed in the past that some students did not include all the code to perform the calculations required to answer the questions in the project. The files often did contain functions that *could* perform the calculations and even contained indications about what the answer should be, but did not actually perform the calculation that would produce the answer. Ensure that your solution does actually do what you intend it to do.

When the marker runs your file, they will check that the answer you claim is actually produced by your code. If your code does not yield the result you claim, then you are likely to lose marks.