CO876 CW2: Practical Report

Budi Arief
24 November 2021
(Based on the original specification by Shujun Li)

For this coursework assignment, you are required to write a practical report, which should include the following two parts:

- Part 1 (**40%**): a log book describing how you did the exercises in the classes that took place in Weeks 17-20 (each week is worth **10%**)
  - The four classes are those on these Fridays: **26 November (Week 17), 3 December (Week 18), 10 December (Week 19),** and **17 December (Week 20)**
- Part 2 (**60%**): a detailed description of **one advanced** exercise you did for **one** selected week (just one out of the four weeks)

This practical report as a whole is worth **20%** of the overall grade for the CO876 module.

You should submit your practical report electronically (as **a single PDF file covering both parts, named [CO876-YourKentID-CW2.pdf]; optionally an additional ZIP file for the source code of Part 2 – see below for more details**) on the Moodle page of this module, by following the relevant link in the "Assessment" section. The submission deadline of the practical report is **Wednesday in Week 25 (i.e. Wednesday 19 January 2022) 23:59pm**.

You can find the more detailed instructions for the two parts below.

Part 1

This part is worth 40% of the coursework assignment (i.e., 8% of the overall grade of the whole module).

For each of the four classes in Weeks 17-20, write a brief description of your work, including how you did the class exercises, what you learned, any difficulties / problems you encountered and how you overcame them (if any).

The marking of this part of the practical report will consider the quality of the work done (as reflected in the documentation) and the quality of the documentation itself including any evidence provided (such as photos or screenshots of your completion of the exercises, as well as your final answers). Demonstrating an understanding of your work is a key component of the practical report and will be a major part of the marking scheme. The marking will be done as follows:

- 0: no attempt or nothing sensible
- 2: poor, attempted but very low quality overall
- 4: mediocre, with some major issues (e.g., major components missing)
- 6: good, with a few major issues (e.g., lack of sufficient evidence of your work, or description on one question is too brief to be considered complete) or many minor issues (e.g., a lot of grammatical errors although the technical content is fine)
- 8: very good, with some minor issues
- 10: excellent, with no or just very few noticeable issues

Notes: The weekly description is supposed to be a relatively light-weight but genuine record of your learning experience in that week's class. Try to be concise but clear about what you want to say. We do not expect each week's summary to exceed 1 A4 side (excluding pictures, tables, diagrams and figures as evidence of your work).

A template for Part 1 is provided below, covering all four weeks (detailed instructions for the first week only).

===== Part 1 Template Starts =====

Week 17 Class

A brief introduction of what this week's class was about, what exercises were given, and what were attempted by you and what you did not attempt and why, both in the class and offline (before or after the class). No need to repeat the detailed class instructions, but just list the exercises briefly as a simple list.

How did I do the exercises and what did I learn about?

Describe how you attempted the exercises and what you learned about from doing them.

What difficulties / problems did I encounter? / What observations and/or thoughts did you have on the exercises?

Describe any difficulties and/or problems you encountered (if any) and how you solved them (or if you could not solve them, explain what you did in your attempt to solve them and why it was not possible to overcome the problems). If you did not encounter any difficulties or problems, state so and describe some observations and/or thoughts you had on the class exercises and results, e.g., how you would design those class exercises, limitations of the results (e.g., they may not be correct or cannot be directly generalised for some other application contexts), any discussions you had with others (e.g., classmates or class supervisors) on the class exercises. Basically, in this section write something beyond the exercises you attempted.

Where relevant, for both the above questions provide evidence of your work (acceptable evidence includes – but not limited to – screenshots, tables, diagrams, pictures of work done on paper).

Week 18 Class

[The same structure as that for Week 17 above.]

Week 19 Class

[The same structure as that for Week 17 above.]

Week 20 Class

[The same structure as that for Week 17 above.]

===== Part 1 Template Ends =====

Part 2

This part is worth 60% of the coursework assignment (i.e., 12% of the overall grade of the whole module).

For this part, you need to write a description of an additional exercise you attempted in a chosen week in Weeks 17-20, which must go beyond the mandatory (basic) exercises in the class of that week. The advanced exercise can be designed by you based on the lecture slides and/or class exercises covered in that week, but can also be one of the optional (advanced) class exercises for that week.

This part must involve the development of some computer programme(s) or script(s) in a **cross-platform** programming language or system of your choice. Here, a cross-platform programming language or system is one allowing the marker to run your code / executable(s) from **all** three mainstream operating systems (Windows, Mac OS, and Linux) ***without*** making ***any*** modifications to your code or executable(s). Python, Java, PHP, Perl, MATLAB, and JavaScript in a client-side HTML page that can run without a web server are good examples. Use an interpreted programming language as much as possible, unless your chosen advanced exercise requires the use of a compiled language like C or C++. When possible, consider using a free online compiler or runtime environment that can be used without the marker registering an account[1] to test the executability of your code and include information about the online compiler / runtime environment you used for the marker to verify your code. If you are not sure about your choice, contact the lecturer for advice.

While you are required to write some programme(s) / script(s), keep it/them as simple as possible, and submit as few files as possible (ideally just one single file so only this one needs executing by the marker). What is important is that the exercise itself should be more advanced, not the programme(s) and script(s) themselves. So if your programme(s) and script(s) turn out to be not very complicated (e.g., below 100 lines of code), it will be fine. As a rule of thumb, your source code should be very easy to test and the source code be easy to read by the marker in just a few minutes after reading your readme file. Comment your code appropriately.

This part should be structured following the template below. Keep this part concise, and please try to limit your work within 5 A4 sides unless you have a lot of screenshots, pictures, tables and diagrams.

===== Part 2 Template Starts =====

Week <*n*> additional exercise

An introduction of which week you chose and why you decided to do the advanced exercise.

What advanced exercise did I do?

Describe the advanced exercise you attempted. If you chose to use one in the class worksheet, explain why you did not create your own one. If you chose to create your own one, explain why it is relevant and how advanced it is compared with other class exercises.

[1] For example, https://www.onlinegdb.com/online_python_compiler for Python, https://www.onlinegdb.com/online_java_compiler for Java, https://www.jdoodle.com/php-online-editor/ for PHP, https://js.do/ for JavaScript, https://www.onlinegdb.com/online_c_compiler for C/C++, https://www.tutorialspoint.com/execute_perl_online.php for Perl, and https://www.tutorialspoint.com/execute_matlab_online.php for MATLAB/Octave.

How did I do the exercise?

Describe what you actually did for the advanced exercise and what you learned about.

What results did I get and what did I learn about?

Describe results you obtained, interpret them against your expectation, and explain what you learned about from doing the exercise.

What went wrong and/or what more could be done?

Describe any issues / mistakes / challenges / problems you experienced, and/or anything additional or more advanced you could do beyond what you attempted.

Source code

Submit your source code (and executable(s) if relevant) as a separate zip file **(named [CO876-YourKentID-CW2-Part2.zip])** or directly copied in the report if the code is simple enough. Include a readme file or a paragraph in this section of this part of the report, explaining how the source code / executable(s) should be run to reproduce results you obtained. In the readme file, give clear instructions on how your code can be executed (ideally using an online compiler) to reproduce the results you reported.

Where relevant, for all the questions provide evidence of your work (acceptable evidence includes – but not limited to – screenshots, tables, diagrams, pictures of work done on paper). You should include enough evidence so that the marker can mark without relying on executing your code/executable(s) and examining the results himself/herself.

===== Part 2 Template Ends =====

The marking scheme of this part is as follows:

- Introduction of Part 2 (5 points)
- What advanced exercise did I do? (10 points)
- How did I do the exercises? (15 points)
- What results did I get and what did I learn? (15 points)
- What went wrong and/or what more could be done? (5 points)
- Source code (10 points – the readme file 5 points, source code running without problems 5 points)

Except for the source code, the marking of other aspects will be based on the correctness of the technical content and the writing quality of the documentation.