

Independent Research Project Report and Code: Guidelines

This document describes some additional guidelines for the report and code, which will help you understand how the independent research projects will be assessed. Please contact Adriana Paluszny (apaluszny@imperial.ac.uk) if you have any further questions.

Essential – You must provide information that identifies you as the author of the code and report.

In the first page of the report, include the following information: University, Course Name, Module Name, First and Last name, Github alias, Email, Date, Title of the project, and Supervisors. If you undertook your IRP as an intern in a company, please also add the company name, address, and company supervisor.

For the code: Add your name/github alias as a comment to each file that you upload to the repository (when possible, in the first line).

REPORT

Prepare your report following the structure of a paper to be submitted to a journal. One possible structure for your document is to follow the guidelines for the journal SoftwareX (<https://www.journals.elsevier.com/softwarex>)

Format - Reports can be prepared with any editor, including MS Word, Latex, OpenOffice, etc, and should be submitted as PDF files. Appendices containing supplementary information can be submitted as separate PDF files; keep in mind that the evaluation will be based on the main submitted document.

Length & Fonts - Reports should be written in font size 11 or 12pt, and should be around 20 pages and not more than 25 pages, excluding references and appendices. While there is no restriction of the font type, when possible, you should prefer a sans-serif font (such as Arial) as recommended by the British Dyslexia Association.

Abstract, Introduction (with Motivation and Problem Statement) (20%). Start with a ~200-word (< 300 word) abstract, and a maximum of six keywords describing your project and computational solution. Your introduction should include a brief summary of the problem that your code/or computational analysis intends to address. Describe your objectives and/or hypotheses, and outline the tasks completed during the independent research project. Describe the state-of-the-art of solutions to the problem, including commercial and academic approaches, and cite these using the reference style described in the “Guide for Authors” document from the SoftwareX journal. Describe briefly the requirements of your solution (Software Requirement Specification – SRS). Clearly state how your independent research project goes beyond the state-of-the-art.

Software Development Life Cycle (SDLC) (25%) - Describe the technical back-end of your solution. Describe if it was developed as a standalone code or if it is an extension of a pre-existing piece of code. In the latter case, briefly describe the ecosystem in which you developed your solution. List and describe what development and operation tools (devops) have been used, and outline the development methodologies used, e.g. Agile, Waterfall, Rational Unified Process (RUP), and describe why. Add an architectural design diagram of your solution. Describe your design rationale and your implementation strategy, including the description of main objects, routines, parallelisation paradigm (shared vs. distributed), and their respective verification and validation routines. Add algorithms, pseudo-code, and if

required also code snippets of the most important functionality that you implemented. Emphasise the novelty and creativity of your work.

Code metadata (5%) - You should succinctly describe the technical platform of your implementation, including compilation requirements and dependencies – such as programming languages and libraries (open source or proprietary), as well as a description of both the implementation and deployment platforms (operating system, basic hardware requirements if any). Add the current code version of your software/computational solution, a link to the code, and links to any developer documentation, user and/or developer support documents.

Implementation & Code (30%) - Describe the simulation capabilities, as well as the test/study cases investigated during your IRP. Describe the unit or system tests, and where applicable also integration tests, that you ran to determine if your program accurately solves the stated problem. Present validation and/or verification results. Describe the implementation of your design, the functionality of your code, and how it may affect accuracy, efficiency and scaling of your computational solution. Present the results of your computational analysis and quantify your results.

Discussion & Conclusions (15%) – Add a brief discussion session to the report. What were the most difficult tasks to resolve in completing the implementation? What are the strengths and limitations of your solution? What are the next steps? What worked and what did not work? Be formal in your analysis, as if you were writing a journal paper, quantify your statements and base your conclusions on findings of your study. If applicable, briefly describe any planned future work related to your project.

A **BIBLIOGRAPHY (5%)** of all resources (books, articles, manuals) that you consulted to understand and develop the project should be included and properly cited in the text.

You may include one or more **APPENDICES** but, note that, these will not be formally assessed.

CODE

Document your uploaded code, and add a readme file that briefly describes all files, how it can be compiled/executed, and what the input/output of your programme is. Your code will be assessed based on the documentation (including installation instructions and user guide), test suites and any additional software infrastructure (e.g., issue tracker/wiki), that have been submitted on GitHub. In addition, markers will take into consideration the implementation and assessment of your code, as described in your written report.