

Faculty of Applied Sciences  
Bachelor of Science in Computing

**COMP490 Final Year Project  
Project Proposal**Academic Year 2022/23

|  |  |
| --- | --- |
| Parsons problem generator and solver | |
|  |  |
| Project number: | Your project number |
| Student ID: | P-19-0834-5 |
| Student Name: | Jane Liu |
|  |  |
| Supervisor: | Philip Lei |
| Assessor: | Your assessor |
|  |  |
| Submission Date: | September 8, 2022 |

Table of Contents

[1 Project Description 2](#_Toc453937400)

[2 Summary of Related Work and Key References 2](#_Toc453937401)

[3 Project Workplan 2](#_Toc453937402)

[4 Risk Assessment 2](#_Toc453937403)

[References 3](#_Toc453937404)

# Project Description

In recent years, with the boom of the computer industry, an increasing number of relevant jobs in this area have been provided and more professional personnel are needed. Consequently, a growing number of people prefer to major in science in computing at their universities in order to find decent jobs after graduating. However, for some of the learners of this program, the courses are not straightforward, especially some courses about improving programming skills. One of the reasons for this tough study of programming is that the students need to practice programming by directly writing codes line by line, which causes a heavy cognitive load for students and makes it easy for students to feel frustrated and even lose interest in science in computing []. Therefore, an advanced method for learning programming, the Parsons problem, will be discussed and proposed in this project.

The aim of this project is to design and build a responsive website for computer and mobile use to generate Parsons problems according to the given original Python code, make students solve the generated Parsons problems, and give feedback for the solutions. The major objectives of this project are as follows:

* Study the existing Parsons problems generators and solvers to get a general idea
* Design system architecture to have a blueprint of the project
* Design and develop a database for storing the account information including the types of users, the generated Parsons problems, the corresponding solution with distractors, the scope tag of problems, and the practice records of students.
* Design and develop a webpage for authorized users to input the original questions with the correct solution and necessary distractors and variants.
* Design and develop a function to generate Parsons problems according to the given questions, solutions, distractors, and variants.
* Design and develop a webpage for students to filter the scope of Parsons problems to practice like introductory programming, data structure and algorithms, and so on.
* Design and develop a webpage to show the generated Parsons problems and let students drag and drop the correct answers to the appropriate positions.
* Design and develop a function to evaluate the code generated by students and give them some feedback
* (Optional) Design and develop a function to show the statistical analysis of the total number of completed practices and the mastery degree of students.

In this project, the Parsons problem is used to assist students in learning programming. This approach allows students just to drag and drop the ready coding blocks separated from the whole original codes instead of writing code directly. In this way, the logic is divided from the syntax [], which has an effect on reducing the cognitive load of students and helping teachers to aim at the specific problems students have []. Besides, with timely feedback, students can have a better understanding of their codes, and be well encouraged to learn further. In addition, different scopes of the concepts covered by the problems can make this project not be limited just to the introductory level of programming study.

There are several challenges in this project. Firstly, it needs to cut the original solution reasonably over all the scopes of the problems. Secondly, it needs to define an appropriate upper bound of the number of distractors to control the complexity of the problems. Thirdly, it needs to develop a 2D view for dragging and dropping the blocks. Lastly, it needs to know the results of students’ code. It may need to run the code on the webpage.

# Summary of Related Work and Key References

This section gives a summary of related work to this project. Key references (e.g. reference books) should also be stated. Use proper citation like [1] to the reference section at the end of this proposal.

# Project Work Plan

This section gives a tentative schedule of how the project work is to be accomplished. Key deliverables (such as key outputs, reports or presentations) should be stated in the schedule. Refer to the FYP calendar for the tentative deadline and dates for reports and presentations.

# Risk Assessment

This section describes the key risks of your project. You are to determine (either quantitatively or qualitatively) the risks related to a concrete situation and a recognized threat to your project. Contingency plans should also be stated.

For example, if you work on a web crawler project in a specific website, there are possibilities that the crawler is forbidden by the website when they discover your intention. If you work on a hardware-related project, there are possibilities that the hardware cannot be delivered on time. Such kind of threats that potentially delay or even fail your project should be pre-cautioned and stated here.

References

[1] Harold Abelson, Gerald Jay Sussman, and Julie Sussman. Structure and Interpretation of Computer Programs. MIT Press, Cambridge, Massachusetts, 1985.

[2] Robert Baumgartner, Georg Gottlob, and Sergio Flesca. Visual information extraction with Lixto. In Proceedings of the 27th International Conference on Very Large Databases, pages 119–128, Rome, Italy, September 2001.Morgan Kaufmann.

[3] Ronald J. Brachman and James G. Schmolze. An overview of the KL-ONE knowledge representation system. Cognitive Science, 9(2):171–216, April–June 1985.

[4] Georg Gottlob, Nicola Leone, and Francesco Scarcello. Hypertree decompositions and tractable queries. Journal of Computer and System Sciences, 64(3):579–627,May 2002.

[5] Georg Gottlob. Complexity results for nonmonotonic logics. Journal of Logic and Computation, 2(3):397–425, June 1992.

[6] Hector J. Levesque. Foundations of a functional approach to knowledge representation. Artificial Intelligence, 23(2):155–212, July 1984.

[7] Hector J. Levesque. A logic of implicit and explicit belief. In Proceedings of the Fourth National Conference on Artificial Intelligence, pages 198–202, Austin, Texas, August 1984. American Association for Artificial Intelligence.

[8] Bernhard Nebel. On the compilability and expressive power of propositional planning formalisms. Journal of Artificial Intelligence Research, 12:271–315, 2000.

[9] Ivan Marsic. A short guide for writing a thesis. http://www.ece.rutgers.edu/~marsic/thesis-guide.html, 2004 [Mar. 6, 2014].

[10] Matlab documentation. http://www.mathworks.com/help/?s\_tid=hp\_ff\_s\_doc [Mar. 6, 2014].

[11] George Sparling. Spacetime is spinorial; new dimensions are timelike. arXiv:gr-qc/0610068v1, 2006.

[12] Ryan Rifkin. Everything old is new again: a fresh look at historical approaches in machine learning. Ph.d thesis, MIT, 2002.