

Course Project

Version 1.7

COMP2090SEF/S209W Data Structures, Algorithms and Problem Solving
COMP8090SEF Data Structures and Algorithms

2026 Spring Semester

Dr. Jimmy S. Ren
Dr. Patrick Chan

1. Introduction

Students of **COMP2090SEF** are required to form a group of 3 (**three**).

Students of **COMP8090SEF and COMP S209W** are required to carry out individual projects.

(1) Task 1: OOP-based application development (50%)

- To develop a Python application with multiple modules/files (no less than 3 modules) using ALL OOP concepts introduced in the course. OOP concepts not covered in the course are also welcome.

(2) Task 2: Self-study on a new data structure AND a new algorithm which are NOT covered in the course (50%)

- To self-study a new data structure **AND** a new algorithm that are NOT covered in the curriculum of this course.

2. Project Description

The project contains **two parts**, and you need to complete both parts:

(1) OOP-based application development (50%)

This project focuses on real-life problem solving with the concept of Object-Oriented Programming (OOP). You have great freedom to choose your topics, but we have some requirements for your project:

The implementation of your work must reflect the concept of OOP. You are required to use ALL OOP concepts introduced in the course. Please refer to the lecture notes and the lecture video recordings (including the extended video lectures on OLE) for Week3 for more details on OOP concepts.

You should aim for an application or a system which can be used in real-life scenarios to solve real problems, such as an inventory system, a library management system, calendar and flight scheduling system, etc. A decent graphical user interface (GUI) may slightly increase your score,

but it is not mandatory for you to have a GUI for your project, as you can present the interface with Python print() function.

Notice: You may extend the project you have done on other courses from the previous semesters with modular programming and data structure added. However, you must

(a) **declare in the final report** that the topic of the project is based on the work (of which group member(s) or of the whole group) from that course e.g. COMP1080SEF, and

(b) **demonstrate in the final report** that your submission for this course contains significant changes and improvements (no less than 60% new content added in) compared with the work submitted for the previous courses.

(2) Self-study on a new data structure AND a new algorithm which are NOT covered in the course (50%)

As an introductory course, it is impossible for us to cover all the data structures and algorithms in one semester. Thus, your self-study will be essential to acquire broader knowledge about algorithm development.

In this course, we will introduce data structures like stack, queue, linked list, and binary search tree, and basic sorting algorithms, such as merge sort, selection sort, bubble sort, and pigeonhole sort. In this part, you are required to self-study one more data structure and one more algorithm (not limited to sorting). Here are possible choices:

a. **Data structure:** binary tree, graph, heap, matrix, etc.

b. **Algorithm:** heap sort, radix sort, shell sort, cocktail sort, etc. (you are not limited to sorting algorithms)

To accomplish this part, each student is required to submit a study report to elaborate on the chosen topic. For your choice of data structure, you need to introduce the abstract data type (ADT) and the possible applications. For your choice of algorithms, you need to analyze the time complexity and give some examples of how the algorithm works.

If your report contains code snippets, a sufficient introduction and explanation are needed to show that you have a good understanding. *Simply pasting the code into the report without any explanation will not be counted toward the grade.* You are encouraged to insert diagrams and tables to illustrate your implementation and running results.

(Continue on next page)

3. Group Forming & Submissions

- For students of **COMP2090SEF**, each project team should have 3 members. **No solo (a group with only one student) is allowed unless you have very good reasons.** Group size larger than 3 is highly not recommended unless you have very good reasons. (I explained the reason why we prefer groups of size 3 during the lecture)
- For students of **COMP8090SEF and COMP S209W**, they should conduct individual projects.
- By **15 February**, the leader of the groups should submit the group name and the list of group members (with student IDs) to the via the following link.

<https://forms.office.com/r/h5pfcae9if>

- By **8 March**, the leader of the groups should conduct a pre-submission of their project via OLE. The pre-submission content should be **your Github repository link** with the following content inside (*please use the same Github repository for both tasks, you may put each task in a separate folder*):
 - Task 1
 - Preliminary code submission
 - Preliminary readme in the frontpage by a README.md markdown fileSee an example Github repo,
(<https://github.com/XPixelGroup/HYPIR>)
 - Task 2
 - Preliminary code submission
 - Preliminary readme in the frontpage by a README.md markdown file
- By **23:59PM, 12 April**, each group need to submit their work (final submission) through OLE in a **zip file**, which contains:
 - Task 1
 - A project report in PDF
(1 cover page + max. 3 pages of TEXT ONLY main content + unlimited pages of appendix with diagrams, tables and others)
 - **Cover page.** Please refer to the requirement on Page 9/10.
 - **Main content** should contain the following parts:
 - Description of the core functions and a summary of the techniques,
 - State and illustrate the usage of the OOP concepts,
 - You may refer to the diagrams and tables in the main text but put the diagrams and tables in the appendix.
 - Declare the use of external resources (packages, other programmers' code, etc.),
 - Self-reflection on the potential weakness and future improvements,
 - Link to your GitHub repository which should contain

- The Python code files
 - User Guide (how to run your Python code)
 - Link to a 5-minute introduction video
 - References.
 - **Appendix** can include figures, tables, flowcharts that cannot be put in the main content due to page limitations, ***declaration of contribution***, project demonstration with test cases, and other necessary contents.
 - Task 2
 - A Study report in PDF
(1 cover page + max. 3 pages of TEXT ONLY main content).
 - **Cover page.** Please refer to the requirement on Page 9/10.
 - **Main content**
 - Introduction of the chosen topics (one new data structure and one new algorithm) and their implementations.
 - You may refer to the diagrams and tables in the main text but put the diagrams and tables in the appendix.
 - Link to your GitHub repository which should contain
 - The Python code files
 - User Guide (how to run your Python code)
 - Link to a 5-minute introduction video
 - References.
 - **Appendix** can include figures, tables, flowcharts that cannot be put in the main content due to page limitations, ***declaration of contribution***, project demonstration with test cases, and other necessary contents.
- (Submit PDF files for both reports instead of OneDrive links)***

- **Important notice:**

- All course project assessments are **competitive** in nature. An average score will be given if the group's work only satisfies the minimum requirements.
- Failing to comply with the requirements may cause score deduction, for example, **late submission and underlength/overlength reports.**
- All the communications must be done through HKMU email for the security reason. Email submission (for group list and proposal) from non-HKMU address will be discarded.
- The submission of final work must be done on OLE. Email submission will not be handled.
- Academic honesty must be respected (refer to Section 6). Any misconduct may result a zero score for the group.

(Continue on next page)

4. Marking Criterias

| Component | Weight | Criteria |
|--|--------|---|
| Problem definition | 10% | <ol style="list-style-type: none">1. Clearly state and formulate the real-life problem to be addressed2. Sufficiently survey how this problem is solved in real life3. State the data structure to be used and explain its usage in the project4. Declare if the project is based on previous work |
| Problem solution (Github pages and Code) | 30% | <ol style="list-style-type: none">1. The program should be runnable and solve the problem in an intended way2. The program should be accurately coded in a professional manner with appropriate comments. A program without any comments will be considered a lack of readability. Score deduction will be resulted.3. The program should present enough programming skills (<i>see project description for details</i>)4. The coding must follow the modular programming concept to define and use classes and functions properly. The project must contain more than one Python file. (<i>see project description for details</i>)5. All OOP concepts introduced in the class must be used in the project. (<i>see project description for details</i>) |
| Conclusion and self-reflection (Group Report) | 30% | <ol style="list-style-type: none">1. The report should comprehensively introduce how the program solves the problem2. Examples, test cases should be provided for demonstration purposes3. Weaknesses of the solution and how to improve should be acknowledged (based on sufficient test cases) |
| Self-study on a new data structure and sorting algorithm (Study Report) | 30% | <ol style="list-style-type: none">1. The report should comprehensively introduce the selected data structure and algorithm. For the data structure, the main methods should be introduced. For the algorithm, the time complexity should be analyzed.2. The development of a new data structure and an algorithm shall be covered.3. Suitable application examples should be presented.4. No score will be given if the report only contains code but has no sufficient explanation. |

The score for the project will be based on the final submission of project work for the team. **Normally, every member will receive the same score on the group report part as the team.** In some cases, a member may be penalized and receive a lower score.

- A team member who does not get involved in teamwork and/or does not respond to communications (through HKMU email accounts) may get a zero score.
- A team may report to me a member who has not responded to team communications for 2 weeks or more. Warning actions will be taken.
- If a team wants to report a free-rider case and request a score deduction on a student, the report or complaint must be made before the final submission of the project. I will not handle late reports or complaints after the submission deadline.

(Continue on next page)

5. Submission and Report Format

The final submission should be made to the OLE. More details will be provided in due course.

Academic reports do not need fancy appearance. Here is a suggested standardized format for your report:

Page size: A4

Font: Times New Roman

Font Size: 12 points

Margin: Normal (2.54 cm all round)

Spacing: Single-line spacing

Please try to follow the above format to write your report.

6. Plagiarism

Plagiarism is using another person's work and pretending it is your own work. In other words, if you are using the ideas of another person without giving credit to the person, it is considered plagiarism. The following are examples of plagiarism:

- Copying content from another team (you are pretending that you have written the content on your own).
- Copying a paragraph from a newspaper without citation (you are pretending that you have written the sentence on your own).
- Using a program from somewhere else in your project without proper reference (you are pretending that you are the author of the program).
- Copying a sentence from a website and making cosmetic changes to the words (you are pretending that you have authored the whole sentence).
- Using a figure or a photo downloaded from a website (you are pretending that you have taken the photo or drawn the figure, it is also a Copyright Infringement issue).
- Using previous work as submission for another credit-bearing assignment is also plagiarism.

Requirement on academic honesty (COMP2090SEF):

1. Each group must make the following declaration on the cover page of the report

Hong Kong Metropolitan University

COMP2090SEF
Data Structures, Algorithms And Problem Solving

Course Project Report
Group 10

Submission date:

We declare that:

- (i) all members of the group have read and checked that all parts of the project (including proposal, code programs, and reports), irrespective of whether they are contributed by individual member or all members as a group, here submitted is original except for source material explicitly acknowledged;
- (ii) the project, in parts or in whole, has not been submitted for more than one purpose without declaration;
- (iii) we are aware of the University's policy and regulations on honesty in academic work and understand the possible consequence when breaching such policy and regulations;
- (iv) we confirm that we have declared in the report about the usage of AI and other generative models, including but not limited to ChatGPT, LLaMA, Gemini, Mistral, and Stable Diffusion, and complied with the instructions provided by HKMU; and
- (v) we are aware that all members of the group should be held responsible and liable to disciplinary actions, irrespective of whether he/she has signed the declaration and whether he/she has contributed, directly or indirectly, to the problematic contents.

We confirm that we have read through and understood the project requirements. We understand that failure to comply with the project requirements will result in score deduction.

| NAME | SID |
|------|-----|
| | |
| | |
| | |

2. The group must acknowledge ALL external resources or code from others referenced. We will only mark the parts that are original, not the parts from others.
3. All members are liable to the plagiarism incidence, even if a free-rider.

Requirement on academic honesty (COMP8090SEF/S209W):

1. Each student must make the following declaration on the cover page of the report

Hong Kong Metropolitan University

COMP S209W/8090SEF
Data Structures, Algorithms And Problem Solving
Data Structures, Algorithms

Course Project Report
Group 10

Submission date:

I declare that:

- (i) I have read and checked that all parts of the project (including proposal, code programs, and reports), they are contributed by me, here submitted is original except for source material explicitly acknowledged;
- (ii) the project, in parts or in whole, has not been submitted for more than one purpose without declaration;
- (iii) I am aware of the University's policy and regulations on honesty in academic work and understand the possible consequence when breaching such policy and regulations;
- (iv) I confirm that I have declared in the report about the usage of AI and other generative models, including but not limited to ChatGPT, LLaMA, Gemini, Mistral, and Stable Diffusion, and complied with the instructions provided by HKMU; and
- (v) I am aware that I should be held responsible and liable to disciplinary actions, irrespective of whether I have signed the declaration and whether I have contributed, directly or indirectly, to the problematic contents.

I confirm that I have read through and understood the project requirements. I understand that failure to comply with the project requirements will result in score deduction.

| NAME | SID |
|------|-----|
| | |

2. Each student must acknowledge ALL external resources or code from others referenced. We will only mark the parts that are original, not the parts from others.