

SENG201 – Software Engineering I

Course Outline

1. Introduction

SENG201 builds on the material introduced in COSC121/131 and COSC122 and prepares students for 300-level software engineering courses (SENG301, SENG302, and SENG365). This course introduces the principles, processes, practices, techniques, and tools of software engineering. The underlying theory and practical applications of a variety of topics in software engineering are covered with particular reference to object-oriented software development in Java. Students who achieve a good grade in the course should be able to:

- Design and implement object-oriented software and its GUI in Java
- Use UML to design and document software
- Learn, use, and evaluate a range of software development practices and tools
- Function effectively in a professional software development environment
- Communicate effectively, in written and oral forms, knowledge and professional opinion

Please consult the UC website for the latest information about the times and locations of SENG201 lectures and labs (the CSSE Department is not responsible for scheduling these). Additionally, most course information will be available via LEARN or on the CSSE and CIS web pages.

2. Course content

The major topics we currently intend to cover fall into the following main themes:

Software engineering principles, processes, and practices: The course will discuss fundamental software engineering principles, types of software engineering activities (e.g., understanding customer requirements and ensuring product quality), and best practices in software development.

Object-oriented programming with Java: Statically typed languages, such as Java, play an important role in software engineering and computing in general. We will use Java to explore differences with dynamically typed languages such as Python and will also cover topics such as object-oriented language features, including inheritance, collections, and events.

Object-oriented software design: UML will be used for object-oriented analysis (modelling the problem domain) and object-oriented design (modelling the solution space). The design process allows increasing detail to be specified in UML until it is sufficient to provide an alternative view to the source code.

Tools and techniques: The software development environment includes a range of tools that support design, coding, testing, and other activities in the software development process. We will focus on individual professional practices that support development in Java; the 300-level course will extend these practices to group work.

Lecture and laboratory material will include parallel coverage of material from these themes. Students are strongly advised to study the recommended reading material to be able to participate fully in classroom discussions and laboratory work. Practical work includes the study of some specific tools and the application of lecture and laboratory material in practical contexts. Students are expected to attend labs regularly and to ensure that they understand fully the material covered. History tells us that failure to complete lab work is likely to be associated with poor grades.

3. Course structure

In addition to attending lectures (three hours per week) and lab sessions (two hours per week), students are expected to devote time each week to preparing for lectures through background reading, contributing to LEARN forum discussions, and consulting the teaching team as appropriate to ensure they fully understand the course material.

Note that material presented in lectures is normally distilled from a variety of sources. Consequently, students should not expect to gain equivalent value by reading a particular section of a single text or from course handouts alone. It is particularly important that students participate fully in and contribute to the lecture process. This includes taking supplementary notes in class, engaging in private study, and discussing study material with tutors.

Model answers will not be provided, partly because of the open-ended nature of 200-level practical work. However, tutors and lecturers will be happy to discuss efforts with students.

4. Assessment

The assessment items for the course are summarized below. Any additions or alterations to assessment or teaching schedules will be advised via LEARN, and students are expected to consult it regularly. Further details about laboratory work will be given as the course progresses.

Item	Description	Weight
Lab test	Typically, a computer-based 90-minute closed-book test.	20%
Project	Typically, a small-scale software development project using JavaFX; done in pairs.	30%
Final exam	Typically, a computer-based 2-hour open-book exam; calculators are not allowed; questions will assume knowledge expected to have been gained through practical work and may be based on examples introduced in labs or the project.	50%

Completing all labs and tutorials will be rewarded with a bonus of 5 marks added to the marks earned in the lab test. Conditions for earning the bonus marks are as follows:

- (1) "All or nothing": All labs and tutorials must be completed to earn 5 marks. There is no partial bonus for completing only some of the labs/tutorials.
- (2) The bonus is based on completing (i.e., a meaningful attempt) and submitting the lab/tutorial. Answers to labs/tutorials will not be marked.
- (3) Each lab/tutorial must be completed two weeks after its release. Labs/tutorials completed later than two weeks after their release will be considered not completed as far as the bonus is concerned, and no bonus will be earned.

- (4) The bonus will be applied after the last lab/tutorial. Since there will be labs/tutorials after the lab test, adjustments to lab test marks will be made after the lab test results are released.
- (5) The total marks of the lab test, including the bonus, cannot exceed 100.

Plusage policy: If the final exam mark is higher than the lab test mark, the final exam mark will replace the lab test mark in the final grade calculation

Students are advised to consult the course materials for details on the policy on drop-dead dates and associated penalties (the department's standard policy may not apply).

Students may apply for special consideration if their performance in an assessment is affected by extenuating circumstances beyond their control. Applications for special consideration should be submitted via the Examinations Office website <https://www.canterbury.ac.nz/study/examinations/> within five days of the assessment. Where an extension may be granted for an assessment, this will be decided by direct application to the course coordinator and an application to the Examinations Office may not be required. Special consideration is not available for items worth less than 10% of the course and/or the following items of assessment: n/a. Students prevented by extenuating circumstances from completing the course after the final date for withdrawing may apply for special consideration for late discontinuation of the course. Applications must be submitted to the Examinations Office within five days of the end of the main examination period for the semester.

The CSSE Department's grading policy states that in order to pass a course you must meet two requirements:

1. You must achieve an average grade of at least 50% over all assessment items.
2. You must achieve an average mark of at least 45% on invigilated assessment items.

If you satisfy both these criteria, your grade will be determined by the following University-wide scale for converting marks to grades: an average mark of 50% is sufficient for a C- grade, an average mark of 55% earns a C grade, 60% earns a C+ grade and so forth. However, if you do not satisfy both the passing criteria you will be given either a D or E grade depending on marks. Marks are sometimes scaled to achieve consistency between courses from year to year.

In the case of an emergency that affects the course, the following provisions apply:

- a) The Course Coordinator may change the nature, number, weighting, and timing of assessments, e.g., tests and examinations may be replaced with assignments of the same weight or different weight at a different time and/or date (which, under certain circumstances, may be outside the prescribed course dates). This may happen without the unanimous agreement of students. An example of such an adjustment might be to delete the examination or postpone it to say the next semester.
- b) Tests and the final exam may move online, in which case the student will be required to install software that validates their identity and which may conduct video surveillance during the assessment. In addition, students may be required to undertake a live video-conference interview to validate the originality of their work.

- c) The Dean may postpone some assessments for up to a year. In such cases, students will be given an initial grade based on whatever assessments have been fielded in the course. The passing grades will not be enumerated but will rather be represented by a P or R grade. The failing grades will be shown as usual. The student may elect to keep the initial grade, or sit the subsequent assessment in which case the relevant numeric grade will be applied. An initial P or R grade will not be downgraded to a fail grade irrespective of the subsequent assessment, except in the case of academic dishonesty. Students who pass the initial grade assessment but fail the second postponed assessment will be given an R grade. At the end of the postponement period, the grades will be fixed as they are, and any blank or indeterminate grades filled with a failing grade (F). Note that the P grade does not count towards GPA.
- d) The 'Special consideration' process will be used for unforeseen circumstances that adversely affect the academic performance of students individually. The usual grounds for this are described in the UC policy 'Special Consideration Procedures and Guidelines', and personal circumstances due to a wider emergency event may also qualify. Students are advised to inform the course coordinator prior to making their application for Special Consideration so they can be informed about potential Special Consideration remedies. The transcript will show 'AEG' if the derivation based on the remedy resulted in a higher grade. Note that special consideration cannot be offered where the combined assessments for consideration exceed 50% of the course assessment weight.

Remedial final exam conditions

1. If a student has a special consideration application for the final exam, the only academic remedy available is for the student to sit a remedial final exam.
2. Remedial final exams are not available outside the special consideration framework.
3. If, at the time of the remedial final exam, your special consideration application has not yet been decided, you are asked to sit the remedial final exam anyway. The grade earned in the remedial final exam will only be considered for your final course grade if the special consideration is approved. If it is declined, your final grade will be based on your grades achieved until and including the final exam (which will count as zero marks if you did not attend it).
4. If you sit both the original and remedial final exams (and your special consideration application is approved), only the remedial final exam mark will be used to determine your final grade.
5. Please contact the course coordinator at the same time that you apply for special consideration, so that we can arrange your remedial final exam.
6. The remedial final exam will take place on campus in the resit week. The resit week is set by the Faculty and happens after the final exam period (detailed time and location information will be available in due course). Please keep this in mind when planning holidays or other activities.

5. Textbooks, recommended reading and reference material

Copies of the course texts [1, 2] are placed on reserve in the Engineering Library (previous editions of the textbooks are also suitable). There is an enormous amount of material available in the university libraries on software engineering topics. Suitable supplementary reading material with general

software engineering coverage includes the books by Pressman and Maxim [3], Sommerville [4], and Pilone and Miles [5]. Additional resource material will be made available as required via LEARN.

- [1] C. Horstmann. Big Java: Early Objects. J. Wiley & Sons, 7th edition (2020)
- [2] B.D. McLaughlin, G. Pollice, and D. West. Head First Object-Oriented Analysis & Design. O'Reilly (2006)
- [3] R.S. Pressman and B. Maxim. Software Engineering: A Practitioner's Approach. McGraw-Hill, 9th edition (2019)
- [4] I. Sommerville. Software Engineering. Pearson Education Ltd., 10th edition (2016)
- [5] D. Pilone and R. Miles. Head First Software Development. O'Reilly Media, 1st edition (2008)

6. Pre-requisites and recommended preparation

Pre-requisites:

- COSC121 or COSC131;
- COSC122;

Some familiarity with background material, particularly programming fundamentals and Python, will be assumed.

7. Teaching staff

Lecturers:

- Mathieu Nassif
mathieu.nassif@canterbury.ac.nz
Jack Erskine 310
- Miguel Morales (course coordinator)
miguel.morales@canterbury.ac.nz
Jack Erskine 210

Tutors:

- Morgan English (senior tutor)
morgan.english@canterbury.ac.nz
Jack Erskine 212
- Danita Sun
tut201@cosc.canterbury.ac.nz
- Hannah Grace
tut201@cosc.canterbury.ac.nz
- Corey Hines
tut201@cosc.canterbury.ac.nz
- Tannin Heely
tut201@cosc.canterbury.ac.nz