COMP1100 — Introduction to Software Innovation

Table of contents

Overview . . . 1
Staff . . . 2
Weekly format . . . 2
Learning outcomes . . . 2
Assessment . . . 3
Competency criteria . . . 4
What the course is about . . . 4
Expectations . . . 4
The project challenge . . . 5
Generative AI policy . . . 5
Seminars . . . 5
Reading the course's sources . . . 5

Overview

Introduction to innovation using computer science and information technology through a discipline-specific team project. Students learn what innovation is, the processes innovators follow, how innovation teams work together, how to make decisions in technology projects, how to use prototyping in the innovation process, and the tools required to deliver and communicate an innovation project.

Semester 2 2025 (28/07/2025 – 22/11/2025), St Lucia, in person, 2 units. This was the **third offering** of the course, with no changes from previous offerings.

- Prerequisite: CSSE1001 or DECO1400.
- Incompatible with COMP7110.
- Jointly taught with **COMP7110** — seminars are joint, studios are separately taught but run similar activities.

The course has its own online notes at comp1100.github.io, organised into numbered chapters that the seminars follow.

Staff

- **Professor Tim Miller** — Professor of Artificial Intelligence, course coordinator — timothy.miller@uq.edu.au
- **Dr Yujun Cai** — Lecturer in Computer Science, course coordinator — yujun.cai@uq.edu.au
- **Dr Ian Clough** — CIS Teaching Associate, Legend

Each team is assigned a **legend** (supervisor) who runs their studio and holds fortnightly one-on-one meetings.

Weekly format

When	What
Monday 5–7pm	Innovation seminar — **weeks 1–5 only**. Attendance strongly encouraged.
Monday or Wednesday	3-hour studio, every week. Attendance **compulsory**.

The studio splits into *first two hours* building key skills for teamwork and innovation, and a *third hour* working on the project. If you can't attend or will be late, tell your team and legend **before** the studio via MS Teams, with evidence.

Because the seminar series stops at week 5, the six seminar notes below cover the entire taught content of the course; everything from week 6 on is studio and project work.

Learning outcomes

After successfully completing this course you should be able to:

1. Work independently as part of a team with diverse backgrounds towards authentic IT innovation solutions.
2. Apply innovation processes to open-ended problems.
3. Collect and analyse data about people and problems.
4. Design and prototype innovation solutions to identified open-ended problems.
5. Communicate problems and solutions to technical and non-technical stakeholders.

Assessment

Competency-based: the whole course is pass/fail, with no contribution to GPA — except that a fail registers as a grade 2. There are four hurdles, and every one of them is team- or group-based and identity-verified.

Assessment	Due	Contents
Studio participation	Weeks 4–9 and 11–13	Participation in weekly seminar and studios, including presentation and practice sessions
Submission 1	4/09/2025, 4:00pm	Team report on customer discovery iteration 1, plus a short individual reflection
Submission 2	25/09/2025, 4:00pm	Team report on customer discovery iteration 2, plus a short individual reflection
Submission 3	13/11/2025, 4:00pm	Team report on iteration 3, an **implemented conceptual prototype**, plus a short individual reflection

Pass = pass all four hurdles. **Fail** = fail one or more.

Submissions are pushed to a branch on the team repository *and* submitted as a zip to the LMS. Each has both a team component and an individual self-reflection, and **you must pass both** to pass the course. Individual contribution to the team component is determined from the **source code repository logs** — which is worth internalising early: the commit history is the evidence.

No safety nets

- **No extensions and no deferrals** on any assessment. 100% late penalty (submissions get a one-hour grace period; studio participation gets none).
- **No supplementary assessment**, because the work is completed progressively as part of a team.
- Studio participation can't be deferred at all — it has to happen in the scheduled timeslot.
- If a team is going to miss a deadline, contact the Course Coordinator *in advance* and be ready to submit whatever is done.

The participation trap

Teaching staff actively monitor individual participation all semester — in studios, in fortnightly one-on-ones, and across collaboration platforms. Students who can't demonstrate active participation and contribution to their team's output **risk failing with the grade capped at 2**, regardless of what the team produced.

Competency criteria

Published at comp1100.github.io/intro.html#competency-criteria.

What the course is about

Five things:

1. Working in a team
2. Giving value to people
3. Solving open-ended problems
4. Thinking creatively
5. Being professional

And why innovation is worth learning in computing: people's requirements are always changing; there's sometimes no clear answer; you need to know and remember *why* you're creating something; and it improves business value.

Expectations

What staff expect of you: attend the seminar and studio every week; actively participate with your team; don't stop others from learning; work hard on the project most weeks; **get out of the building and talk to potential users**; stay open-minded; be flexible and self-directed; be curious; behave professionally and ethically; learn stuff and have fun.

What you can expect of staff: attend and give the seminars and studios; set goals or activities for studios each week; pointers when you're stuck; advice on the process to follow; regular positive and negative feedback on deliverables and weekly contributions; help navigating to realistic solutions; genuine interest in your project.

Classroom rules: laptops and mobiles closed unless working on a seminar task (notes and slides are online — take notes by hand if needed); no talking over others; no messy food or drinks; phones on silent; participate. *"There are no stupid questions, stupid answers, or stupid people."*

The project challenge

> Improving the student experience at UQ (and beyond?)

Teams choose one or two challenges from comp1100.github.io/intro.html#project-challenges to work on for the semester.

Generative AI policy

Generative AI may be used **for anything except generating interview or other data**. The point of the course is that the data comes from real people — see customer-discovery[1].

Seminars

- 2025-07-28-introduction-to-innovation[2] — Week 1
- 2025-08-04-business-model-canvas[3] — Week 2
- 2025-08-11-experimentation-and-customer-relationships[4] — Week 3
- 2025-08-18-minimum-viable-products-and-prototyping[5] — Week 4
- 2025-08-25-the-history-of-computing[6] — Week 5a
- 2025-08-25-prompt-engineering[7] — Week 5b

Reading the course's sources

The seminars draw on four books:

- **Tina Seelig**, *inGenius: A Crash Course on Creativity*
- **Steve Blank**, *The Four Steps to the Epiphany* and *The Startup Owner's Manual*
- **Alexander Osterwalder**, *Business Model Generation* (Strategyzer)
- **Eric Ries**, *The Lean Startup*

[1] courses/comp1100/customer-discovery.pdf
[2] courses/comp1100/2025-07-28-introduction-to-innovation.pdf
[3] courses/comp1100/2025-08-04-business-model-canvas.pdf
[4] courses/comp1100/2025-08-11-experimentation-and-customer-relationships.pdf
[5] courses/comp1100/2025-08-18-minimum-viable-products-and-prototyping.pdf
[6] courses/comp1100/2025-08-25-the-history-of-computing.pdf
[7] courses/comp1100/2025-08-25-prompt-engineering.pdf