Lecture 0 — Admin

CITS2005 Object Oriented Programming

Department of Computer Science and Software Engineering
University of Western Australia

Welcome to CITS2005

Unit Coordinator: Dr Andrew "Gozz" Gozzard

Standard rule in my lectures: Raise your hand to ask a question at any time.

Shout out to Dr Max Ward who developed almost all the material for this unit.

Unit

This unit is intended to develop your understanding of Object-Oriented Programming concepts in general, via Java as a specific tool.

We tend to focus on *understanding* concepts and being able to reason about them, not just memorizing them.

The learning outcomes for this unit are for you to be able to

1. implement object-oriented design of solutions to real-world problems;
2. design Java programs using packages and classes with inheritance, generics, interfaces and abstract classes;
3. understand the use of multithreading for designing Java programs with concurrency; and
4. make effective use of software development practices to write, test, debug and document Java programs.

Unit Philosophy

A philosophy of this unit is that learning a programming language is like learning a natural language. You must immerse yourself to become fluent.

We will be seeing a lot of code examples in lectures. We will be running lots of code live in the lectures.

Please take this to heart when it comes to the labs and revision. The only way to write good code is to write lots of bad code first.

Contact

Unit Coordinator: Dr Andrew "Gozz" Gozzard

Office hours available on LMS (may change across semester)

MS Teams Team: "Preferred" place for (publicly shareable) questions and discussion

Unit email: `cits2005-csse@uwa.edu.au`

Please use public discussions so others can benefit.
Keep unit email for non-public questions or contact that must be directed to staff.
Never use my personal emails for unit-related contact.

Classes

Lectures

- Mondays 14:00-15:00
- Wednesdays 10:00-11:00
- Monday 03/03 (next Monday) is a public holiday: No lecture
- Wednesday 19/03 is Prosh: No required attendance, lecture will still run and be recorded

Labs

- Work on lab tasks where you can get guidance from lab facilitators
- See timetable
- Starting week 2
- May attend multiple, but you should have an assigned time if crowded

Resources

- LMS (`https://lms.uwa.edu.au/`) should contain all required links and resources
- Content will be released as we go, typically a week in advance
- The textbook is available via LMS
- Announcements will be made on LMS
  - You are expected to be aware of these announcements

Assessments

Tests (In-person, Invigilated, BYOD)

- Formative Test (0%): Thursday 27/03 (week 5)
- Test 1 (20%): Thursday 10/04 (week 7)
- Test 2 (20%): Monday 05/05 (week 10)

Project (20%, Take-home)

- Released Thursday 08/05 (week 10)
- Due Thursday 22/05 (week 12)

Exam (40%)

See Unit Outline on LMS, note "Academic Information" section is out of date.
Details, including samples when appropriate, will be released ahead of time on LMS.

Labs

Labs are not assessed, but are a core part of the content.

You are expected to do them, and if you do not you will likely struggle with the content.

In previous years, students who did not do the labs often failed the exam.

Generative AI

If you have not, please read (Using AI Tools at UWA: A Guide for Students).

- Limited use permitted as "educational and study tools"
  - I advise you to not use it as anything more than a search engine, as previous students have undermined their own understanding while being conned into thinking they are learning
- May only be used in assessment where it is **explicitly permitted** (it is not in this unit)
- Any (permitted) use must still be clearly cited as an external resource to not be considered plagiarism
- I do not want to have to give people misconduct strikes for trying to circumvent assessments (again)

I personally avoid using these tools for learning entirely, as the evidence and my personal experience indicates that while it may *feel* like it is helping you learn, it is actually undermining you, and it only increases the performance of below-average programmers, without them actually improving.