

SWEN20003

Object Oriented Software Development

Subject Introduction

Bach Le

bach.le@unimelb.edu.au

University of Melbourne

© University of Melbourne 2023

Teaching Staff

- Dr Bach Le (bach.le@unimelb.edu.au)
- Dr Andrew Valentine (andrew.valentine@unimelb.edu.au)

Teaching Staff

- Dr Bach Le (bach.le@unimelb.edu.au)
- Dr Andrew Valentine (andrew.valentine@unimelb.edu.au)

Head Tutor:

- Tharun Dharmawickrema (t.dharmawickrema@unimelb.edu.au)

Tutors:

- Stella Li
- Catherine Muir
- Charlie Ding
- Yifan Guang
- Chenghao (Lee) Li
- Risa Pais
- Hao Xu

A bit about myself

- Bach Le

- ▶ Lecturer in the School of Computing and Information Systems
- ▶ Australian Research Council Fellow (ARC DECRA Fellow)

- Education

- ▶ Post doctoral researcher at Carnegie Mellon University, California, USA.
Topic: Automated Software Testing
- ▶ PhD in Software Engineering - Singapore Management University, Singapore.
Topic: Automated Software Bug Fixing.
- ▶ B.Sc. in Information Technology - Hanoi University of Science and Technology, Vietnam. **Topic: Automated Software Verification.**

A bit about myself

- Academic Experience

- ▶ Academic at the University of Melbourne since 2019
- ▶ Teaching
 - ★ Software Engineering and Computer Science (SWEN20003, COMP30026, etc)
- ▶ Research
 - ★ Data analytics (mining software repositories)
 - ★ Automated Software Debugging: testing, bug fixing, verification

Learning Outcomes - Handbook

On completion of this subject you are expected to:

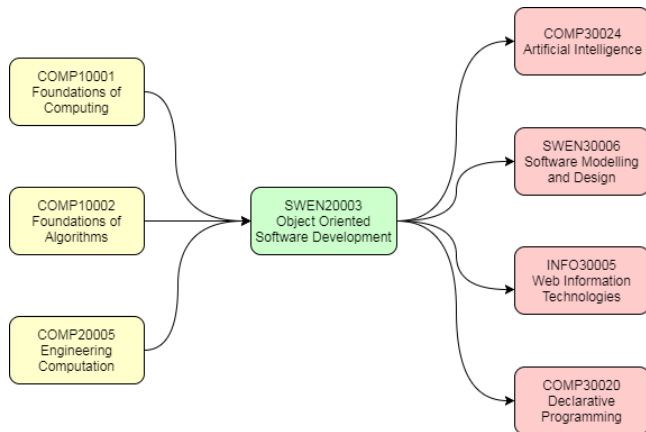
- Apply software design principles to object-oriented design
- Develop object-oriented models for a medium-sized software system
- Evaluate design trade-off of different designs
- Implement an object-oriented design in a suitable language
- Use commonly available object-oriented design frameworks for application development
- Apply knowledge of basic science and engineering fundamentals

Learning Outcomes - In Simple Language

On completion of this subject you will be able to:

- *Write* **Object Oriented** programs in Java
- *Use* an Integrated Development Environment (IDE) to develop software
- *Abstract* a problem specification the Object Oriented way
- *Design* a software solution, not just write it
- *Apply* programming techniques, frameworks, and conventions, to other Object Oriented languages

OOSD in Context



Lecture Schedule

SWEN20003 Object Oriented Software Development - Semester 1, 2023					
Block	Week	Date	Lecture Topic	Assessment	Workshops
Java Foundations OOP Fundamentals	1	February 27, 2023	Subject Introduction		
		March 2, 2023	Java - A Quick Tour		
	2	March 6, 2023	Classes and Objects 1		Grok IDE
		March 9, 2023	Classes and Objects 2		
	3	March 13, 2023	Classes and Objects 3		Classes and Objects
		March 16, 2023	Software Tools/Bagel		
	4	March 20, 2023	Arrays and Strings		Classes and Objects with Git
		March 23, 2023	Input and Output		
	5	March 27, 2023	Inheritance 1		Bagel Introduction
		March 30, 2023	Inheritance 2		
	6	March 31, 2023		Project 1 Released (16.30 pm)	
		April 3, 2023	Interfaces		Inheritance
		April 6, 2023	Revision		
		April 7, 2023		Project 1 Initial Due (16.30 pm)	
		Non Teaching Week			
Advanced OOP and Software Design	7	April 17, 2023	Mid Semester Test		Inheritance and Interfaces
		April 20, 2023	Class Diagrams		
		April 21, 2023		Project 1 Due (16.30 pm)	
		April 21, 2023		Project 2 Released (16.30 pm)	
	8	April 24, 2023	Generics		Class Diagrams
		April 27, 2023	Collections and Maps		
	9	May 1, 2023	Design Patterns 1	Project 2A Due (16.30 pm)	Generics/Collections/Maps
		May 4, 2023	Design Patterns 2		
	10	May 5, 2023		Project 1 Marks Released	
		May 8, 2023	Exceptions		
		May 11, 2023	Software Testing and Design		Design Patterns
		May 12, 2023		Project 2A Marks Released	
	11	May 15, 2023	Asynchronous Programming		Exceptions/Testing
		May 18, 2023	Advanced Java Concepts		
	12	May 19, 2023		Project 2B Due (16.30 pm)	
		May 22, 2023	Revision		Event Driven/Advanced
		May 25, 2023	Wrapup - Exam		
				Project 2B Marks Released	
Last Updated: 20-02-2023					

Assessment Breakdown

Project 1 - Initial	0% (-3% for no submission)
Project 1	8%
Project 2A	8%
Project 2B	14%
Mid-semester Test	10%
Workshops	5%
Weekly Quizzes	5%
Final Exam	50%

Hurdle:

To pass the subject, students must obtain at least 50% overall, 15/30 in project work, and 30/60 in the mid-semester test and end-of-semester written examination combined.

Assessment Dates and Deadlines

- Project 1 - Initial - Due - 16:30pm, Friday April 7th (Week 6)
- Mid-semester Test - Monday April 17th (Week 7), TBA.
 - Project 1 - Due - 16:30pm, Friday April 21st (Week 7)
 - Project 2A - Due - 16:30pm, Monday May 1st (Week 9)
 - Project 2B - Due - 16:30pm, Friday May 19th (Week 11)
 - Workshops - Assessed at the end of each workshop
- Weekly Quizzes - Due on Sunday . e.g. week 1 quiz is due on the 5th of March

Workshops

Each week there will be a 2 hour workshop:

- Workshops start in **week 2**
- Workshops contain practical exercises that give you hands-on experience on concepts covered in the lectures
- Doing these exercises will be important for you to understand the concepts, do your projects and final examination
- Workshops will be assessed: one mark will be assigned for each workshop, which will count 5% towards the final mark.

Assessment - Workshops

- Workshops will be assessed:
 - ▶ students who complete workshop exercises and keep up with the subject topics generally perform well in the subject
 - ▶ previous semesters student feedback highlighted the lack of interactions with other students due to shifting to on line mode
- Each workshop is assigned 1 mark
- To obtain the 1 mark assigned for a workshop you must:
 - ▶ attempt the pre-workshop question before attending the workshop, and show your work to the demonstrator during the workshop (the solution does not have to be complete but a reasonable attempt is acceptable);
 - ▶ join a small group the demonstrator will assign you to during the workshop - in your group (or sometimes individually), work on a question selected by the workshop demonstrator and show a satisfactory solution (or a reasonable attempt).
- The final workshop mark will be the sum of the **best eight workshop marks** earned during the semester weighted to 5%

Assessment - Weekly Quizzes

- Each week there will be a Canvas quiz on the weekly lecture content.
- The quiz will test your understanding of the concepts introduced in the lectures.
- Each quiz will have a maximum of 10 questions, and is expected to take less than 20 minutes.
- Each weekly quiz is due on Sunday 11.59 pm of that particular week (e.g. quiz for week 1 will be due on Sunday 6th March 11.59 pm) - students will get a single attempt to complete the weekly quiz.
- The final quiz mark will be the sum of the **best eight marks earned from quizzes** during the semester weighted to 5% (each quizz will have an equal contribution towards the final quiz mark).
- The first week quiz will be a practice, and will not be assessed; although you will see a mark for it, the final mark will be based on the best eight out of the 11 quizzes, starting week 2.

Textbook

- The subject content will be based off Absolute Java by Walter Savitch. You can buy the textbook (4th edition or higher) if you wish, but this is **not required**.

Development Tools

- Any new language has a learning curve
- To reduce the impact, we'll be using Grok initially to give you programming practice
- Starting from week 2, we will introduce IntelliJ, an Integrated Development Environment (IDE)
- You will need to use an IDE for the Projects

Academic Misconduct

- Work with friends if you like on **workshop** questions
- All **assessed** work (projects) is to be done by you alone
- You can discuss overall approach to solving problems with peers or others
- **Do not** show your code to peers, in person or electronically, or ask peers for code
- When in doubt, ask lecturer or tutor
- Sophisticated software is used to identify cheating
- Cheating is grounds for disciplinary action; repeat offence is grounds for expulsion
- See <https://academichonesty.unimelb.edu.au/>

Student Representatives

The first two expressions of interest via email to me (bach.le@unimelb.edu.au) will be selected as student representatives.

Extra Resources

Practice (and exam-like) problems:

- Hackerrank
- Codecademy
- Codesignal

Grok

- Relevant practice materials are now available on Grok
- The materials teach content to *complement* and *reinforce* the lectures
- Go at your own pace
- Not assessed, does not contribute to your marks
- Don't just answer the questions; it is assumed you will have **read the slides** as well
- Most of the code should be intuitive, or at least make some sense
- **On Canvas go to Modules, scroll down to Week 1, and then click Grok Learning, then click on "Grok Start Here" to register with Grok - if you do not do this you will not be able to start using Grok.**

Lectures

- Lecture Delivery Plan
 - ▶ Weeks 1-6: Bach Le
 - ▶ Weeks 7-12: Andrew Valentine
- During the first week we will give you a quick tour of the Java foundations
 - ▶ We will only spend the first week on the content in this topic - you are expected to read this material and familiarize
 - ▶ We assume you have experience in **at least one** programming language (Python and/or C)
- We will teach you Object Oriented Programming concepts throughout the semester, enabling you to write good **Object Oriented Programs in Java**

Lecture Delivery Modes

- In the lecture theatre
 - ▶ Lectures will be recorded.

How to Better Engage in the Subject

- Attend lectures and participate in quizzes, activities and discussions
 - ▶ Anecdotal evidence shows that students who attend lectures do better in subjects

How to Better Engage in the Subject

- Attend lectures and participate in quizzes, activities and discussions
 - ▶ Anecdotal evidence shows that students who attend lectures do better in subjects
- Attend the workshop you are enrolled in and participate
 - ▶ Your demonstrators are the best resource for **specific** questions, particularly regarding the projects

How to Better Engage in the Subject

- Attend lectures and participate in quizzes, activities and discussions
 - ▶ Anecdotal evidence shows that students who attend lectures do better in subjects
- Attend the workshop you are enrolled in and participate
 - ▶ Your demonstrators are the best resource for **specific** questions, particularly regarding the projects
- Be active on subject forums

How to Better Engage in the Subject

- Attend lectures and participate in quizzes, activities and discussions
 - ▶ Anecdotal evidence shows that students who attend lectures do better in subjects
- Attend the workshop you are enrolled in and participate
 - ▶ Your demonstrators are the best resource for **specific** questions, particularly regarding the projects
- Be active on subject forums
- Practice programming -*practice makes you at least close to perfect!*

How to Better Engage in the Subject

- Attend lectures and participate in quizzes, activities and discussions
 - ▶ Anecdotal evidence shows that students who attend lectures do better in subjects
- Attend the workshop you are enrolled in and participate
 - ▶ Your demonstrators are the best resource for **specific** questions, particularly regarding the projects
- Be active on subject forums
- Practice programming -*practice makes you at least close to perfect!*
- Ask questions, **lots** of questions