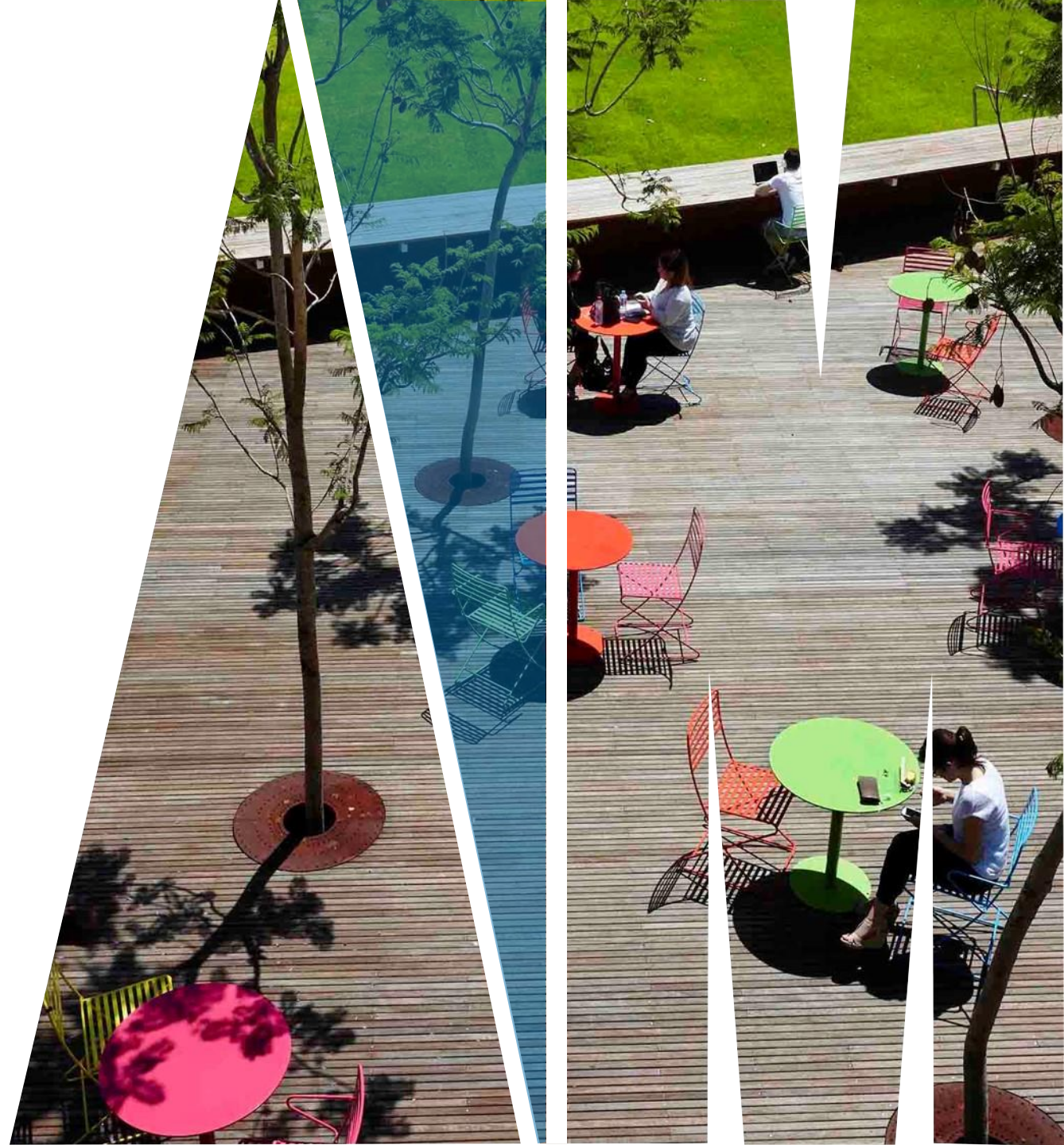


FIT2099 Object-Oriented Design and Implementation

Overview of FIT2099



Acknowledgement of Country

**I would like to acknowledge the Traditional Owners
of the land on which I am today,
the Yalukit Willam clan of the Boon Wurrung People.**

**I would like us to pay our respects to their Elders past, present and emerging.
We acknowledge and respect their continuing
relationship to the lands upon which we meet.**

LEARNING OBJECTIVES

- 1- **Iteratively construct object-oriented designs** for small to medium-size software systems, and describe these designs using standard software engineering notations including UML class diagrams (in conceptual and concrete forms), UML interaction diagrams and, if applicable, UML state diagrams;
- 2- **Evaluate the quality of object-oriented software designs**, both in terms of meeting user requirements and in terms of good design principles, using appropriate domain vocabulary to do so;
- 3- **Implement object-oriented designs in an object-oriented programming language (i.e., Java)**, using object-oriented programming constructs such as classes, inheritance, abstract classes, and generics as appropriate;
- 4- Use available language tools, such as debuggers and profilers, and good programming practice to **debug the implementations systematically and efficiently**;
- 5- **Use software engineering tools** including UML drawing tools, integrated development environments, and revision control to create, edit, and manage artifacts created during the development process.

HOW FIT2099 works

Class (lecture) and labs

Class 1 hour/week (Weeks 1-12)

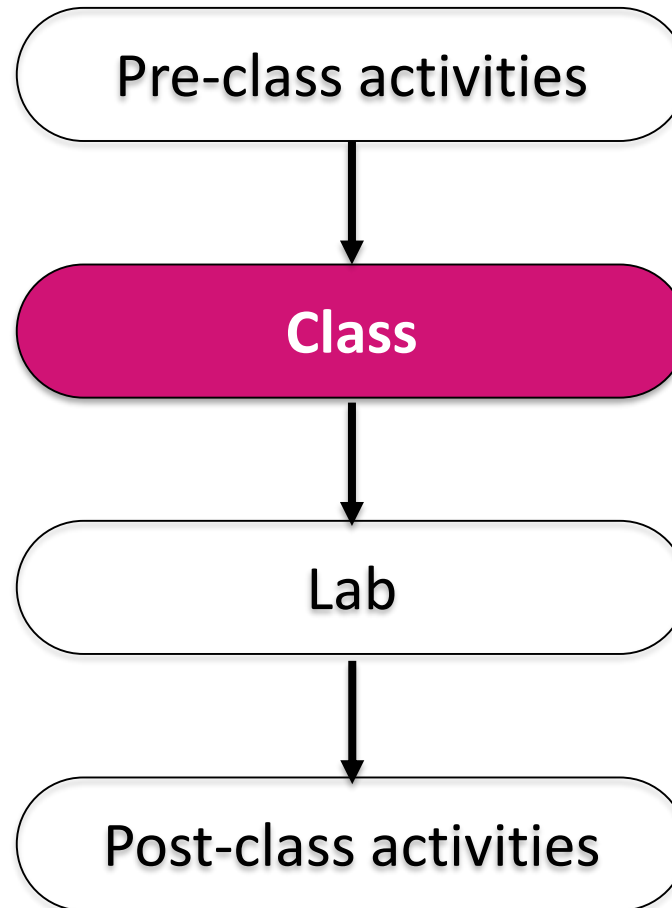
Labs 3 hours/week (Weeks 1-12)

Labs start Week 1, assessed labs in weeks 2-5

Independent study

- Set **pre-class and post class activities** on EdLessons (these are required)
- **Assignment** project in teams

THE WEEKLY WORKFLOW



THE ASSESSMENT

Bootcamp held in **labs** and **EdLessons** – 10%

3 **Assignments** done in teams (same for each assignment)

- Assignment 1 – 20%. Due Friday Week 6
- Assignment 2 – 20%. Due Friday Week 8
- Assignment 3 – 20%. Due Friday Week 11

Assignments involve designing and implementing extensions to an existing object-oriented system

Final **eExam** – 30%

THE BOOTCAMP



PART A (in the labs weeks 2-5) - 8%

- Weekly Java programming activities which include Object-Oriented principles.
- The cover the first five weeks of the semester during the labs.
- Activities in weeks 2-5 are assessed in the labs.



PART B (EdLessons, weeks 1-12) - 2%

- These are intended to help you boost your Java and OO implementation skills in preparation for the Assignments, in addition to the labs.
- They are to be completed online, at your own pace by the end of Week 12.
- You will spend around 3 hours per week depending on your previous experience with Java

THE BOOTCAMP (in labs, weeks 1-5)

GO TO EdLessons – Week 1-5



Bootcamp in the lab

1

Attempt the bootcamp BEFORE the lab

2

Get feedback / ask questions to TAs during the lab

3

You can keep working and updating your repository



4

Commit everything before your next lab (a week after)

5

Go to your handover interview with your TA in your next lab
(marking and final feedback on **completion** and **quality** of the work)

THE BOOTCAMP (in labs, weeks 1-5)

Marking

- Based on a **rubric for each week**
- The rubric considers:
 - **quality** of design and implementation work,
 - **alignment** between design and implementation and
 - the **handover interview**
- The rubric will be available via Moodle
- Marks per Bootcamp:

Week 1	Week 2	Week 3	Week 4	Week 5
0 marks	1 mark	2 marks	2 marks	3 marks

THE BOOTCAMP (in EdLessons)

GO TO EdLessons

Week 0 is optional
(recommended if you are
new to Java programming)



Week 0 (highly encouraged if you are new to Java programming)



- Java for Beginners - Part 1 (~60 minutes)

Week 1

Assessed
completion only



- Pre-class Activities: Java for Beginners - Part 2 (~45 minutes)



- Pre-class Activities: Video lessons (~50 minutes)



- Class activities (~50 minutes)



- **Bootcamp in the lab**



- Post-class Activities (~30 minutes)



- Supplementary Materials

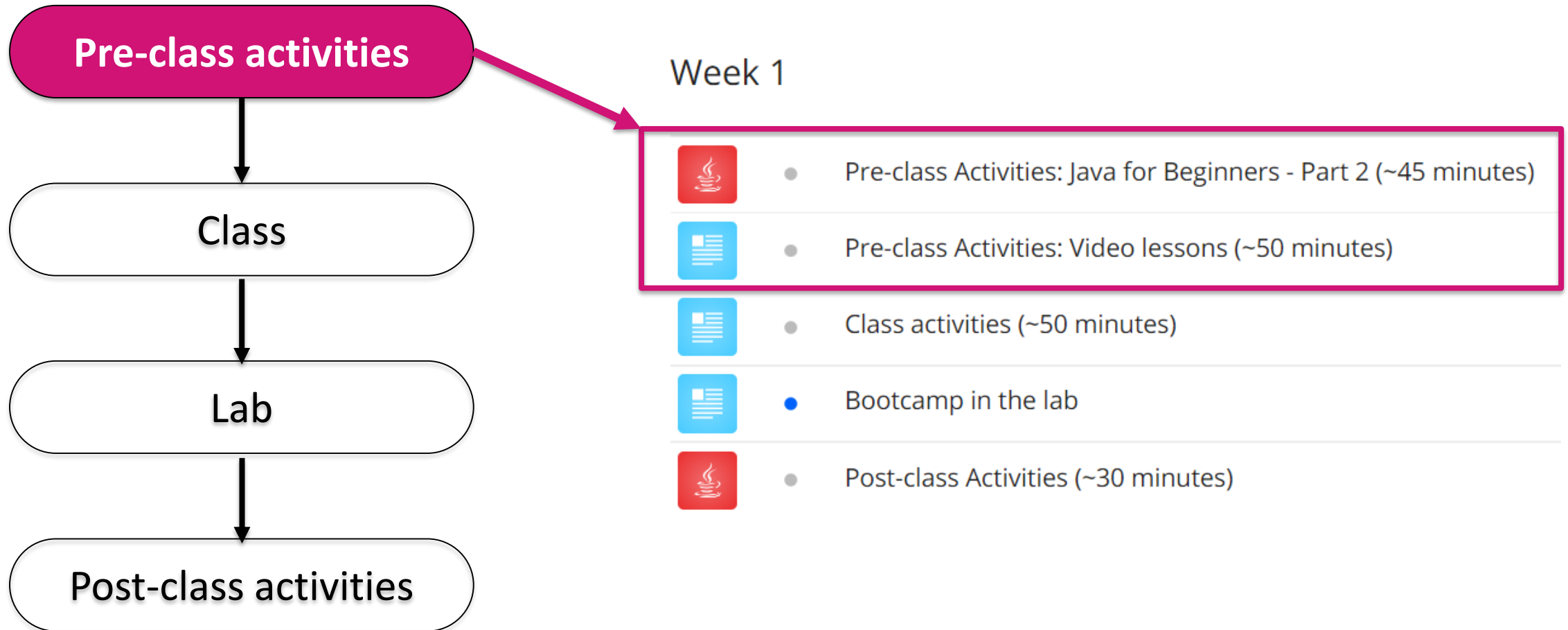


- [Optional][Assignment Support] 🎮 Game Development Concepts

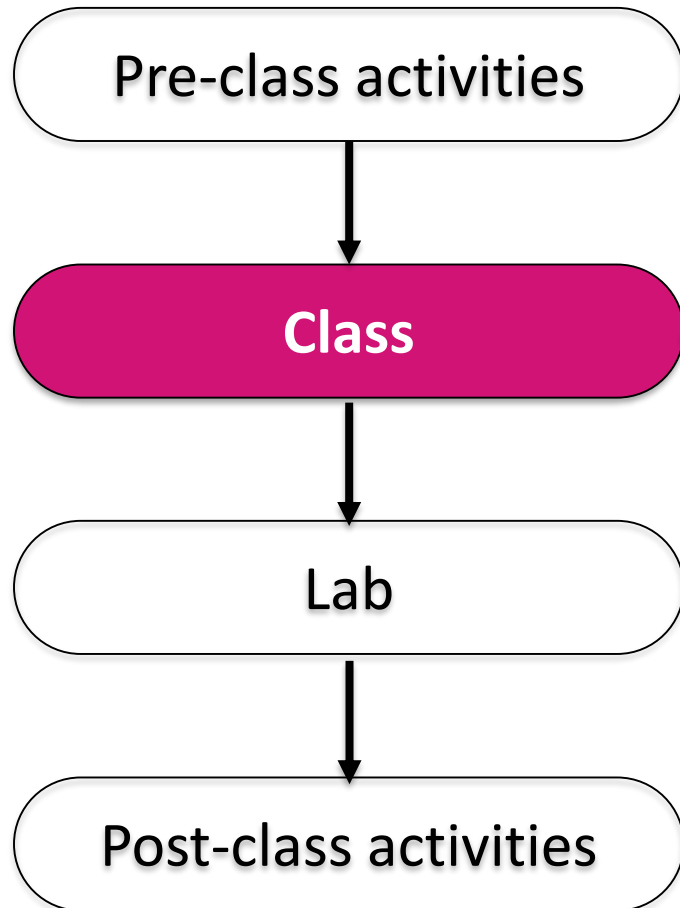
Additional resources
not assessed



THE WEEKLY WORKFLOW



THE WEEKLY WORKFLOW



Week 1



- Pre-class Activities: Java for Beginners - Part 2 (~45 minutes)



- Pre-class Activities: Video lessons (~50 minutes)



- Class activities (~50 minutes)

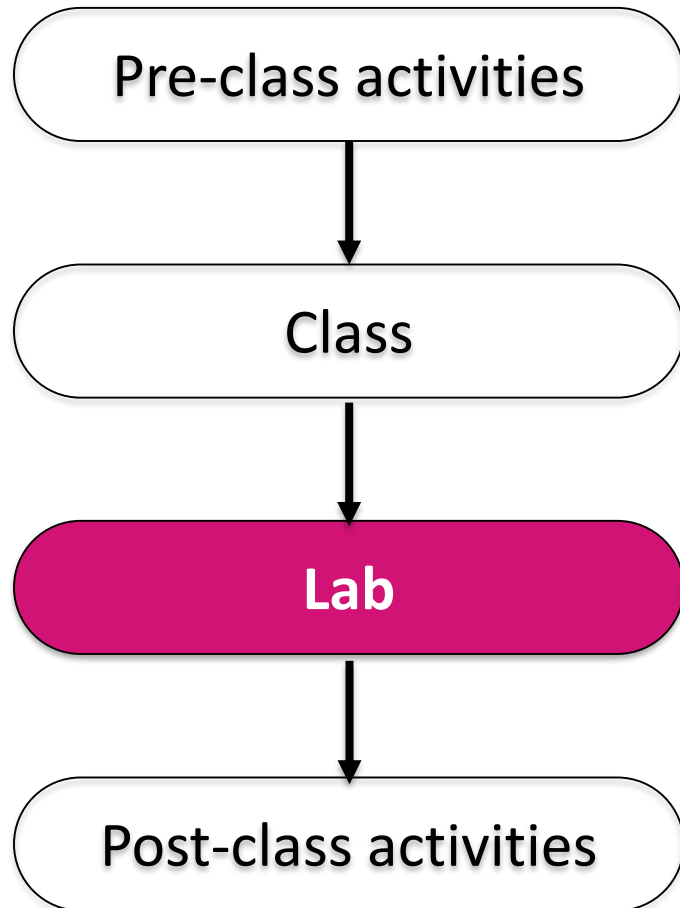


- Bootcamp in the lab



- Post-class Activities (~30 minutes)

THE WEEKLY WORKFLOW



Week 1



- Pre-class Activities: Java for Beginners - Part 2 (~45 minutes)



- Pre-class Activities: Video lessons (~50 minutes)



- Class activities (~50 minutes)

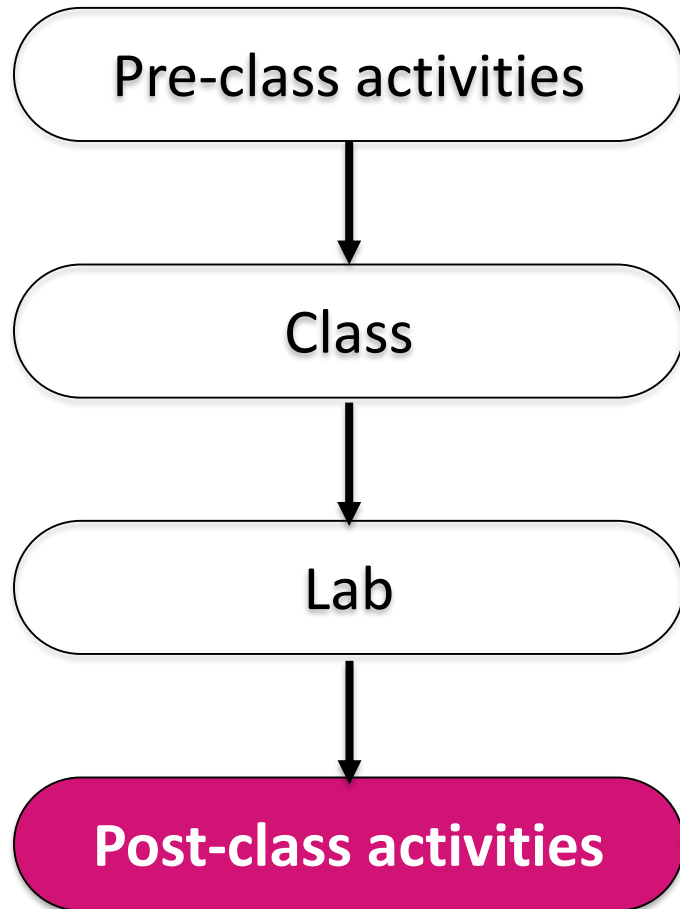


- Bootcamp in the lab



- Post-class Activities (~30 minutes)

THE WEEKLY WORKFLOW



Week 1



- Pre-class Activities: Java for Beginners - Part 2 (~45 minutes)



- Pre-class Activities: Video lessons (~50 minutes)



- Class activities (~50 minutes)



- Bootcamp in the lab



- Post-class Activities (~30 minutes)

THE BOOTCAMP (in EdLessons, weeks 0-5)

Marking

Completion % by Week 12 x 2 marks = Mark for Bootcamp Part B

For example:

75% x 2 marks = 1.5 marks

THE PROJECT (Assignments 1, 2 and 3)



Project (in labs, Weeks 5-12)


- Done in teams to give you practice at communicating with your peers about design
- It is one project split into three submission points
- The idea is to keep the workload steady rather than have a mad rush at the end
- The **Design** is at least as important as **Implementation**
 - even if you get it 100% working, you can still fail
 - for a good mark your code must be maintainable, extensible, and exhibit other signs of good OO design practice
- Several feedback opportunities
- All the team members are responsible for the whole project, rather than individual parts

THE PROJECT (Assignments 1, 2 and 3)

Please, read the **assignment rules document** for more details

GO TO EdLessons

Assignments



- Assignment Rules (+ Survival Kit) ✕
- Assignment 1: Design
- Assignment 2: Develop
- Assignment 3: Further Design and Implementation

THE PROJECT (Assignments 1, 2 and 3)

Marking

- Based on a **rubric for each assignment**
- The rubric considers both **completeness and quality** of work in terms of Design AND/OR Implementation
- The **handover interview** is part of the assignment (the week after the deadline)
- The rubrics will be available via Moodle

THE FINAL EXAM

- Closed book (you will not need to memorise much)
- **Practical** (design and implementation exercises)
- Example exams will be made available via Moodle
- Two classes in Weeks 11 and 12 focused on the Exam
- **Thirty marks** in total

WHAT SOFTWARE WILL BE NEEDED?

You need a working Java development environment to work on labs and assignments at home

- we suggest JDK15; links are available on EdLessons

You will use a git repository to manage all project data for the assignments

- An **individual** repository and a **team** repository will be assigned to each of you and your team for the labs and assignments, respectively.
- You will need a git client
- Most modern IDEs have one integrated (including **IntelliJ** which we will be using and supporting in this unit)
- if you learned GitKraken in other units and want to use it, we won't stop you (but we probably won't be able to support you if it breaks)

GETTING FEEDBACK

- **In the labs**, ask your TA
 - Oral feedback and written feedback (along with the summative assessment)
 - If you're at Clayton, there are two per class, you're encouraged to ask either one at any time
 - You will get direct feedback during each handover interview
- Feedback in **ED Discussion Forum**
 - If you ask something you may get feedback from other students and teaching staff
 - Questions sent by email that are not of a personal nature, but about the unit content in general, will be redirected to Ed Discussion
- Come to a **consultation session**
 - these will be organized when availabilities are known and when demand becomes clear, from Week 2.
 - You can attend ANY consultation session! Bring your questions or assignment drafts.
- Come to the **class!**
 - You can ask questions to the lecturer in turn about the pre/post class lessons you have previously watched or followed by coding along



MONASH
University

Thanks



MONASH
University

