

1.0 - Week 1 - Workshop (MA)

Learning Objectives

- Find out about the unit and how things will work.
- Get familiar with ed and zoom.

Week 1 Padlet Discussion Board link: <https://monashmalaysia.padlet.org/fermi/2022week1>

What is a workshop?

In this unit's workshop, we

- apply, broaden and deepen our understanding of the concepts covered in the Moodle lesson. You absolutely need to have prepared the previous lesson in order to follow and make the most of it!
- ask questions on padlet or zoom or ed, which the tutors will answer or escalate to the lecturer.
- gently point out when the lecturers are mistaken.

If you do not attend the workshop, a recording will be made available, typically later in the same day. You can then work on the workshop on your own time and at your own pace. An ed lesson better suited for asynchronous use will also be made available. Do what works best for you.

Getting familiar with the Moodle site

If you have not done so already:

1. Go to the [Moodle site](#) of this unit
2. Review the [Unit Information page](#).
3. Check that you have access to the videos in [Week 0](#).
4. Check the [important Documents](#).

Workload expectations

Every week:

- Go over and understand the Moodle lesson: **2.5 hours**
- Attend and participate in the workshop: **2 hours**
- Attend and participate in the applied session: **2 hours**

For the assessments, over four weeks:

- Work on your assignment tasks: estimated at **12 hours** but very variable. Start early and submit on time!
- Attend the interview: 15 minutes. During the interview, your tutor will ask questions to assess your understanding and will provide feedback.

Passing the unit

The official stuff

The in-semester assessments of the unit include the 3 major assignments, which are worth 20%, 40%, and 40% of the total mark, respectively. In all these assignments, you will work in a group of peers and submit your group work by a specified deadline.

Your work will be marked for correctness and quality as well as for the division of work in your group. Also, note that a **crucial component** of each assessment is the post-submission interview, where your knowledge and contribution to the group work will be assessed by a tutor.

Note that this time the unit **does not** have an end-of-semester exam, i.e. your final mark will be determined solely by the accumulated mark you get in the assessments. In order to pass, your unit total must be over 50%, or 50 out of 100 marks. You will also need to get a minimum of 45% in A2 and A3 in semester assignments as these assignments are hurdles. A1, however, does not contain a hurdle.

Stern and Serious Reminders

- The reason you are enrolled in this unit is to learn, not to pass the unit.
- The 3 in-semester assessments are not only to evaluate you but also to help you learn.
- Remember that each of the assignments has a relatively high weight. Try to put decent effort into each of them.
- At first glance, it may look easy enough to cheat or cheese the 3 assessments. But be warned: in this case, you won't learn and you most likely will fail the interviews.

Getting help

If you need help with the unit, we recommend you try this first.

- Ask your tutors and classmates during applied sessions.
- Go to a consultation. See [times on Moodle](#).
- Ask on ed. There should be staff answering Monday-Friday.
- For admin-related questions, email muhammad.fermipasha@monash.edu.

Getting familiar with ed threads

1. First, get familiar with the [rules and guidelines](#) of ed, if you haven't already done so.
2. Create a thread on the [discussion board](#). If you do not have a question or anything related to the unit to share, you can simply make a post under "Social" (e.g. introduce yourself). In any case, *use the proper category!*
3. Finally, browse ed and post a reply on at least two threads.

[Offline students: do pretty much the same]

Getting familiar with breakout rooms

We will create breakout rooms, randomly assign you to them, and ask you to:

1. Unmute yourself,
2. Get out of bed and turn on your camera,
3. Introduce yourself and share your favourite crêpe topping (or something more relevant to you),
4. Continue the previous activity, review other students' memes and shenanigans, and make your own.

[Offline students: you can stay in bed]

Getting familiar with ed challenges

In order to get familiar with ed, write a python function that returns True if and only if the argument is even.

Complexity of function below

Consider the function below.

```
def below(n:int) -> int:
    sum = 0
    for i in range(1, n+1):
        for j in range(1, i+1):
            sum += i
    return sum
```

Question 1 *Submitted Jul 25th 2022 at 10:58:29 am*

What is the runtime complexity of the function below?

- ☐ $O(0)$
- ☐ $O(1)$
- ☐ $O(n)$
- ☐ $O(n \log n)$
- ☒ $O(n^2)$
- ☐ $O(n^3)$
- ☐ $O(2^n)$

Question 2 *Submitted Jul 25th 2022 at 11:03:44 am*

Run by hand the function below for $n = 4$. What does it output?

30

Breakout room problem solving!

Each of the next two activities will be run in breakout rooms. We will reconvene after some time to solve each of them.

[Offline students: try to solve them too!]

First Improvement on function below

Consider the function below.

```
def below(n:int) -> int:
    sum = 0
    for i in range(1, n+1):
        for j in range(1, i+1):
            sum += i
    return sum
```

Write a function that outputs the same value as this function, but that has a $O(n)$ time complexity (we suppose that arithmetic operations are $O(1)$). Note that this is not an implementation trick. You probably will have to do some thinking! In order to pass this question, your function not only has to generate the same output as the function below, but it also has to run faster! If you get the error that your program ran for too long, than it most likely does not have the right time complexity (even though you may have improved it).

Hint:

Second Improvement on function below

Consider the function below.

```
def below(n:int) -> int:
    sum = 0
    for i in range(1, n+1):
        for j in range(1, i+1):
            sum += i
    return sum
```

Write a function that outputs the same value as this function, but that has a $O(1)$ time complexity (we suppose that arithmetic operations are $O(1)$). Note that this is not an implementation trick. You probably will have to do some thinking! In order to pass this question, your function not only has to generate the same output as the function below, but it also has to run faster! If you get the error that your program ran for too long, than it most likely does not have the right time complexity (even though you may have improved it).

Hint:

Feedback Form

Weekly Workshop Feedback Form

Question 1 *Submitted Jul 24th 2022 at 9:43:30 pm*

I am enrolled in:

☐ 🇦🇺 Australia

☒ 🇲🇾 Malaysia

Question 2

What needs improvement?

No response

Question 3

What worked best?

No response

Question 4

How engaged were you by the workshop?

☐ 🇸🇸 Very engaged

☐ 🇸🇸 Engaged

☐ 😐 Not impressed

☐ 😐👁️👁️👁️ Lost