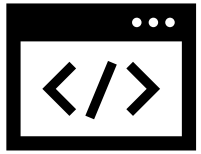
IMPERIAL



BIOE50010 – Programming 2

Computer Lab 10: Assignment Q&A

Binghuan Li, Maria Portela, Gauthier Boeshertz, Samuel George-White, Yilin Sun, Kamrul Hasan, Wenhao Ding, Siyu Mu, Lito Chatzidavari

8 December, 2025

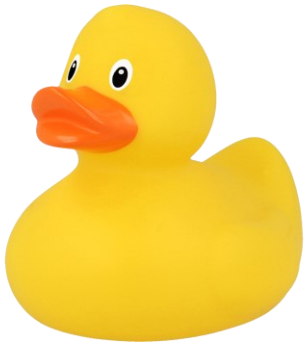
N.B. – What's Next?

- Assignment week **Q&A sessions**
  - Monday 9:00 -11:00 AM, RSM G08
  - Thursday 12:00 – 6:00 PM, RSM 338
- Questions outside the classroom: **Ed Discussion**
- Deadline for the assignment: **9:00 AM on Friday 12 December 2025**
- Questions will not be answered **after 9:00 AM on Thursday 11 December 2025**

- A revision session has been scheduled on **Monday 5 January 2026, 1:00 - 2:00 PM,** at **RSM 131.** (*Thoughts?*)

- Live coding exam on **Friday 9 January 2026, 10:00 AM - 12:00 PM**

Assignment Myth-busting



Rubber duck debugging

- **Assumptions are assumptions are assumptions.**
  - "*Will you crush my code with an irregular-shaped board ?*" No.
- **You can use additional methods,** *e.g.* helper functions.
  - However, additional methods will not be assessed directly.
  - They must not replace or conflict with the behaviours of the required methods.
  - Additional functionalities *may* be a bonus, but not a must.
  - "*Will I be penalised for the compensated efficiency?*" Time it yourself.
- **Efficiency is only meaningful if you compare using the same setup.**
  - … and it will only be measured if your code already works correctly.
- **"*The assignment is unit-tested, so readability does not matter*."**
  - Not True! Think this as how would you teach us to reproduce your brilliant work!

Questions?

Appendix 1: **General Good Coding Practice**

- Code is read much more often than it is written. Code should always be written in a way that promotes readability.
- PEP 8 provides coding style guide for Python programming from the authors' perspectives. Key advice to summarise:
  1. Use **intention-revealing**, **descriptive** names
  2. Adhere to the **proper code layout** (*e.g.*, use consistent 4-space indentation)
  3. Keep **comments**, but good comments do not excuse unclear code

Identifier Type	Example Name	Naming Convention
variables	`playBoard`	Lower Camel Case
functions	`displayBoard`	Lower Camel Case
	`display_board`	Snake Case
classes	`BioengPerson`	Upper Camel Case
constants	`MAX_CAPACITY`	Constant Case

Avoid using names *e.g.*,

- `myList` ← vague
- `data1` ← "noisy"
- `l`, `O` ← 'l' or '1'? 'O' or '0'?

Appendix 2: **Documenting a Function**

argument annotation: a and b are `float`

return annotation: c is `float`

```
def calculate_pythagoras(a: float, b: float) -> float:
    """
    Calculate the hypotenuse of a right-angled triangle.

    Args:
        a (float): Length of side a.
        b (float): Length of side b.

    Returns:
        float: Length of the hypotenuse.

    Example:
        >>> calculate_pythagoras(3, 4)
        5.0
    """
    c = (a**2 + b**2)**0.5
    return c
```

consistent 4-space indentation

Documentation strings:

- Function description
- Arguments
- Return
- Example usage

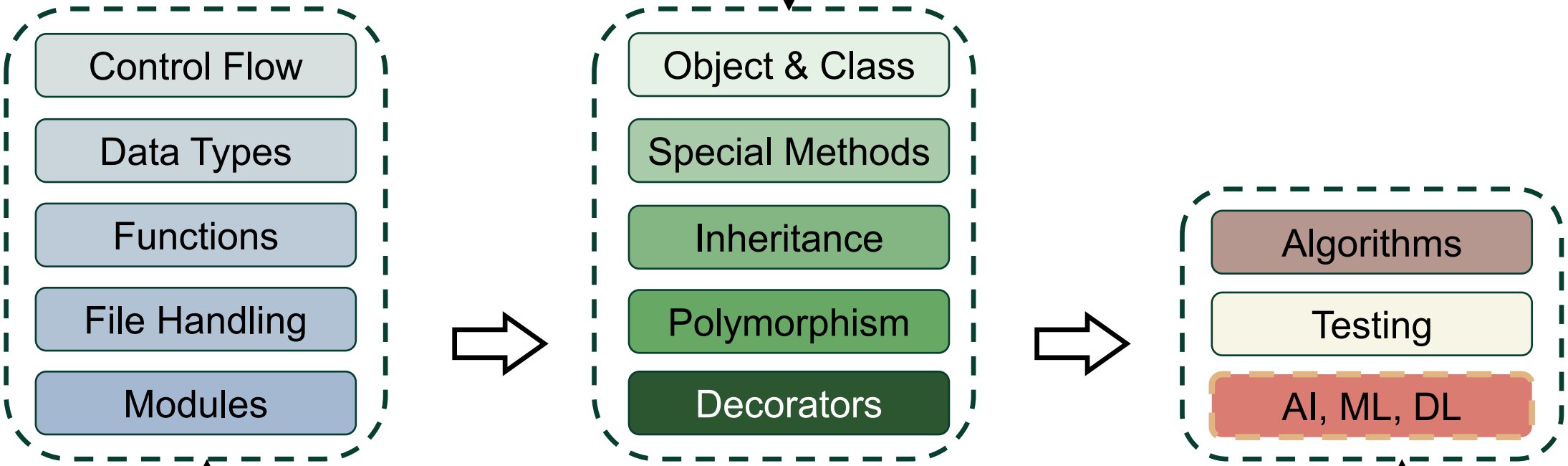
IMPERIAL

Debriefing of Programming 2 Labs

8 December, 2025

Programming 2

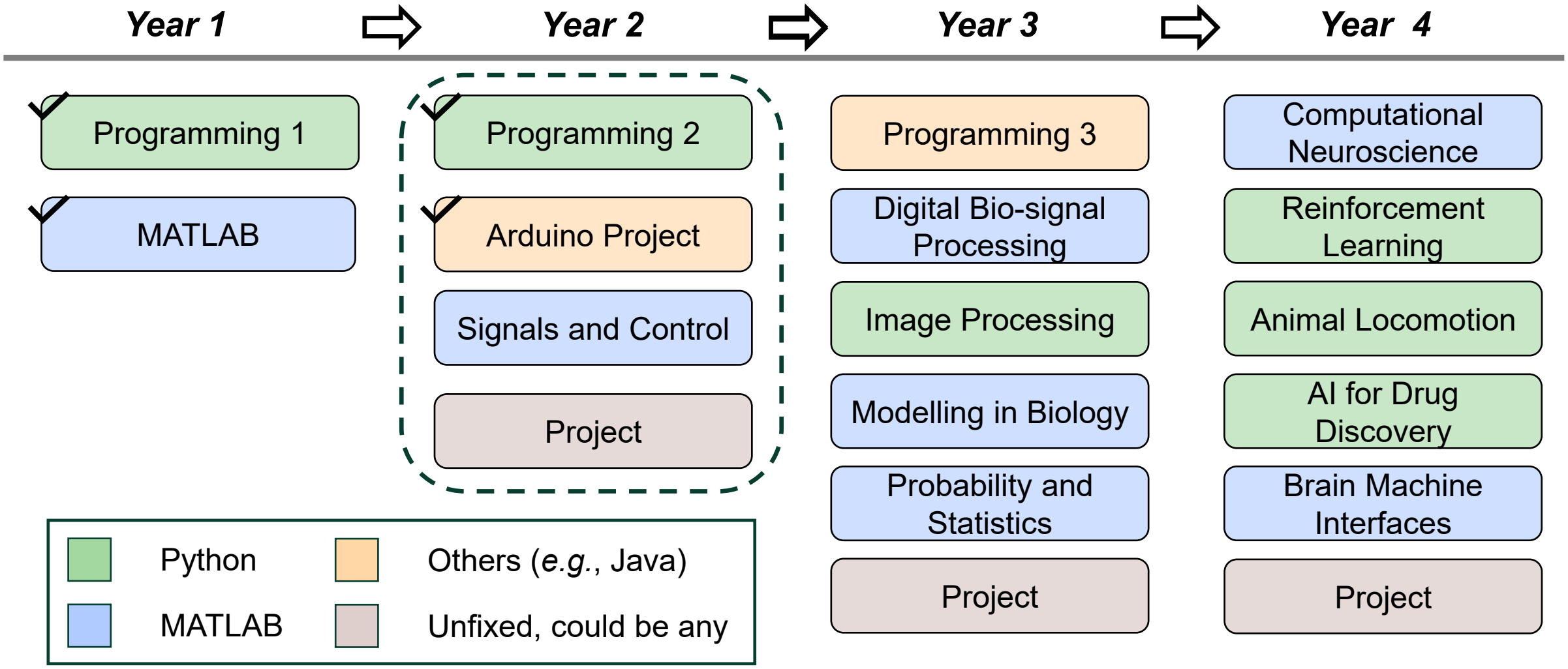
Focus:
- Design data structures using classes.
- Apply relationships through inheritance and polymorphism.
- Enhance designs with decorators and special methods.



Focus:
- Grasp Python syntax and logic.
- Use data types, control flow, and functions.
- Apply modules and file handling.

Focus:
- Implement and test efficient algorithms.
- Evaluate code performance and reliability.
- Recognize the role of AL/ML in research.

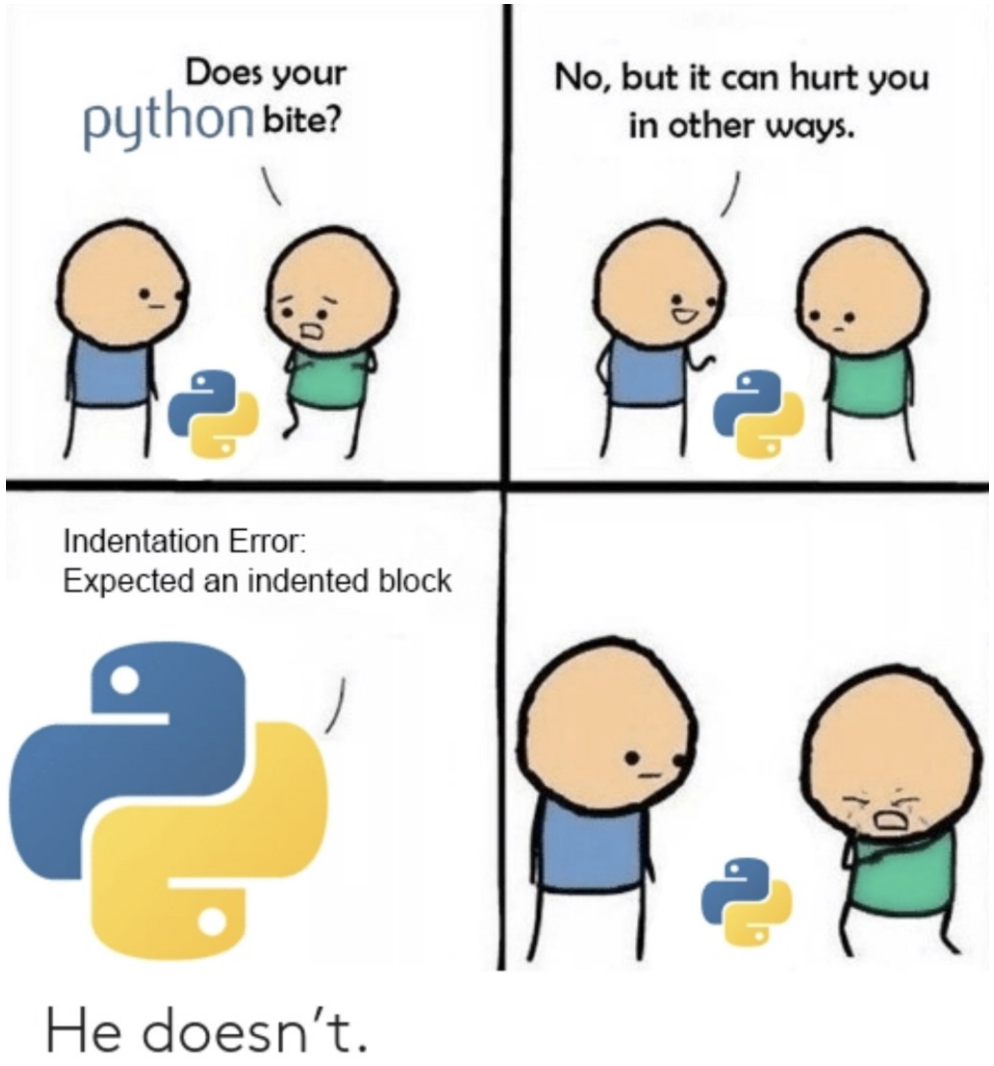
Beyond Programming 2



* Information retrieved from the Module Descriptor 2025-26. Indictive only.

Exam

- Do not just look at code and say “oh, that makes sense”.
- Do write the code from a blank page.
- The syntax of every line of code must make sense to you.
- The algorithm that’s being implemented needs to make sense to you.
- Make sure you are familiar with **Python IDLE interface** before exam.



The End }

Bioeng Coding Beats on Spotify

- Have assignment questions? We're on Ed Discussion until 9:00 AM on Thursday 11 December 2025
- Beyond this module…
- If your coding life does not go on hiatus after this course, the *real* fun is yet to come.
- GTA feedback on SOLE… we hope we've earned your nice words! 🥺
- Until next time! **Merry Christmas!**

14629.202110

BIOE50010 - Programming 2 2021-2022

Open

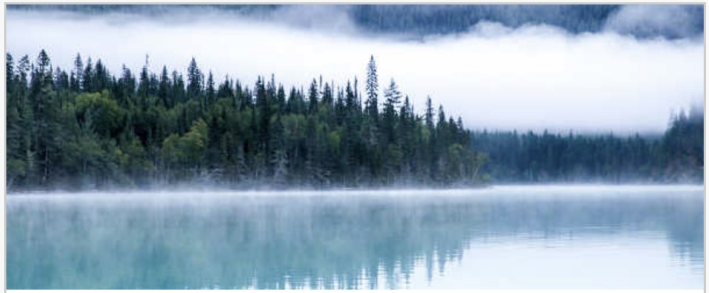
Multiple Instructors

14629.202210

BIOE50010 - Programming 2 2022-2023

Open

Multiple Instructors



14629.202310

BIOE50010 - Programming 2 2023-2024

Open

Multiple Instructors

14629.202410

BIOE50010 - Programming 2 2024-2025

Open

Multiple Instructors

14629.202510

BIOE50010 - Programming 2 2025-2026

Open

Multiple Instructors

More years to follow…