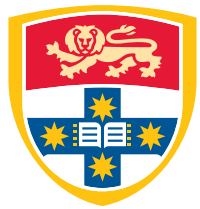
# Introduction to Programming (Adv)

School of Computer Science, University of Sydney



# Copyright Warning

# Lecture 1: Introduction to Programming

*Course overview and general guidelines*

This unit is *INFO1910: Introduction to Programming Advanced*.

This will require students to study both standard (INFO1110) as well as advanced course contents.

This is not a stand alone document for INFO1910. Advanced students will also need to review the lecture notes from the standard lecture as well as the UoS page here.

Special consideration and academic integrity are the same for all students in the University.

Take note of late penalties that apply to advanced.

# About this unit

This unit is the lead-in to ALL the technical units in the School of Computer Science

Admission is only gained through departmental permission via Sydney Student.

This unit has pre-requisites.

- Every student must participate in the entrance examination being a programming test.

- AND one of the following:
  - *ATAR sufficient to enter Dalyell program*
    OR
  - *Demonstrated programming ability from industry standpoint*

# About The Coordinator/Lecturer

The unit coordinator and lecturer is
**Dr. John Stavrakakis**

PhD in Computer Science
*Specialises in 3D computer graphics*

Overall course administration and design

Please be considerate of his time.

# About the Tutors

Richard McKenzie

Programming, mathematics, outdoorsmanship, weightlifting and gaming

Simon Dowd

An aspirational Lego Botanist

# Delivery format

13 weeks of standard lectures

13 weeks of 2 hour lab that you should attend.

Assessments are different to standard

As advanced students, it is expected that you are actively reading, studying and practising throughout the course. It is very easy to fall behind in this unit of study.

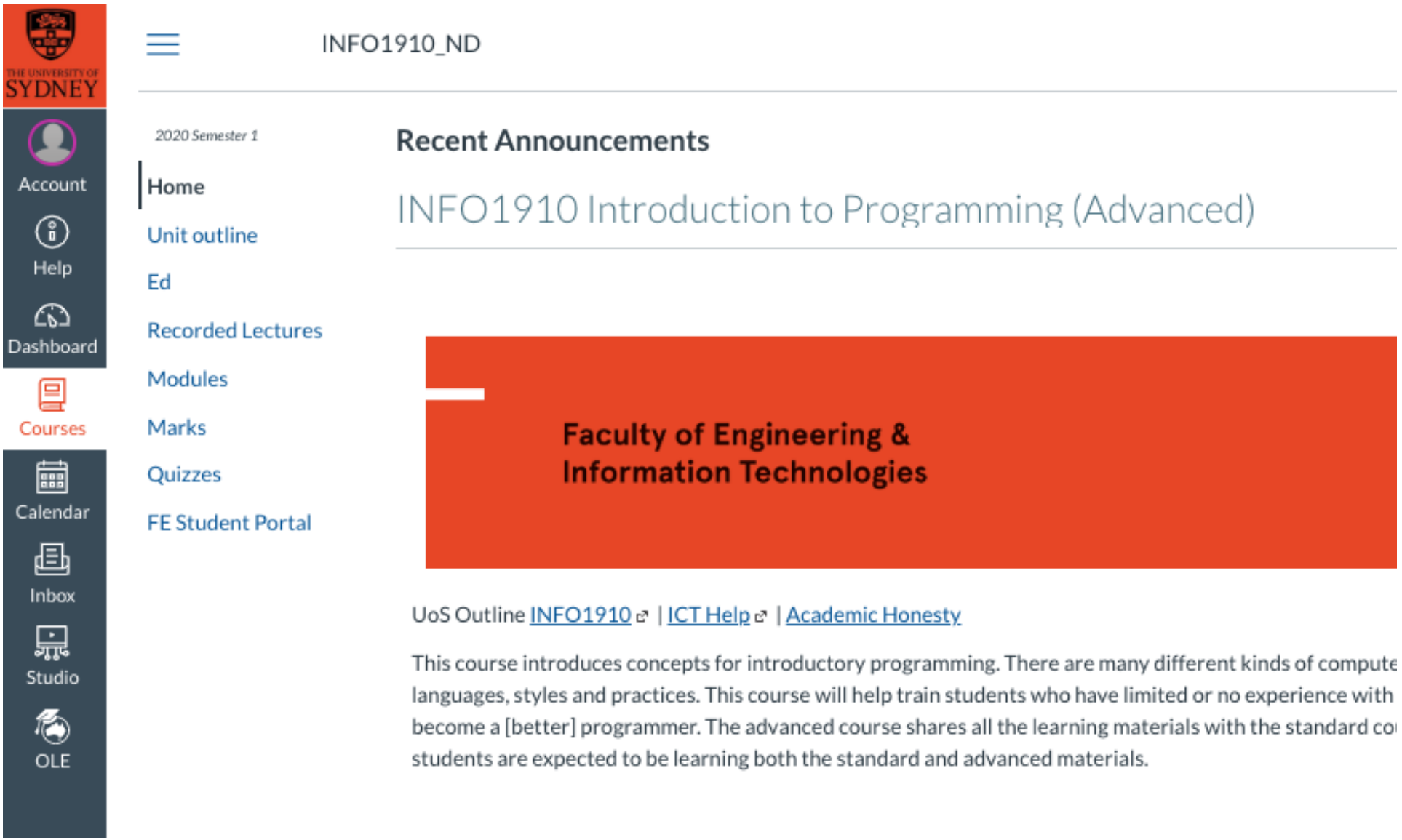
# Online resources: Canvas

This semester Canvas will be used for:

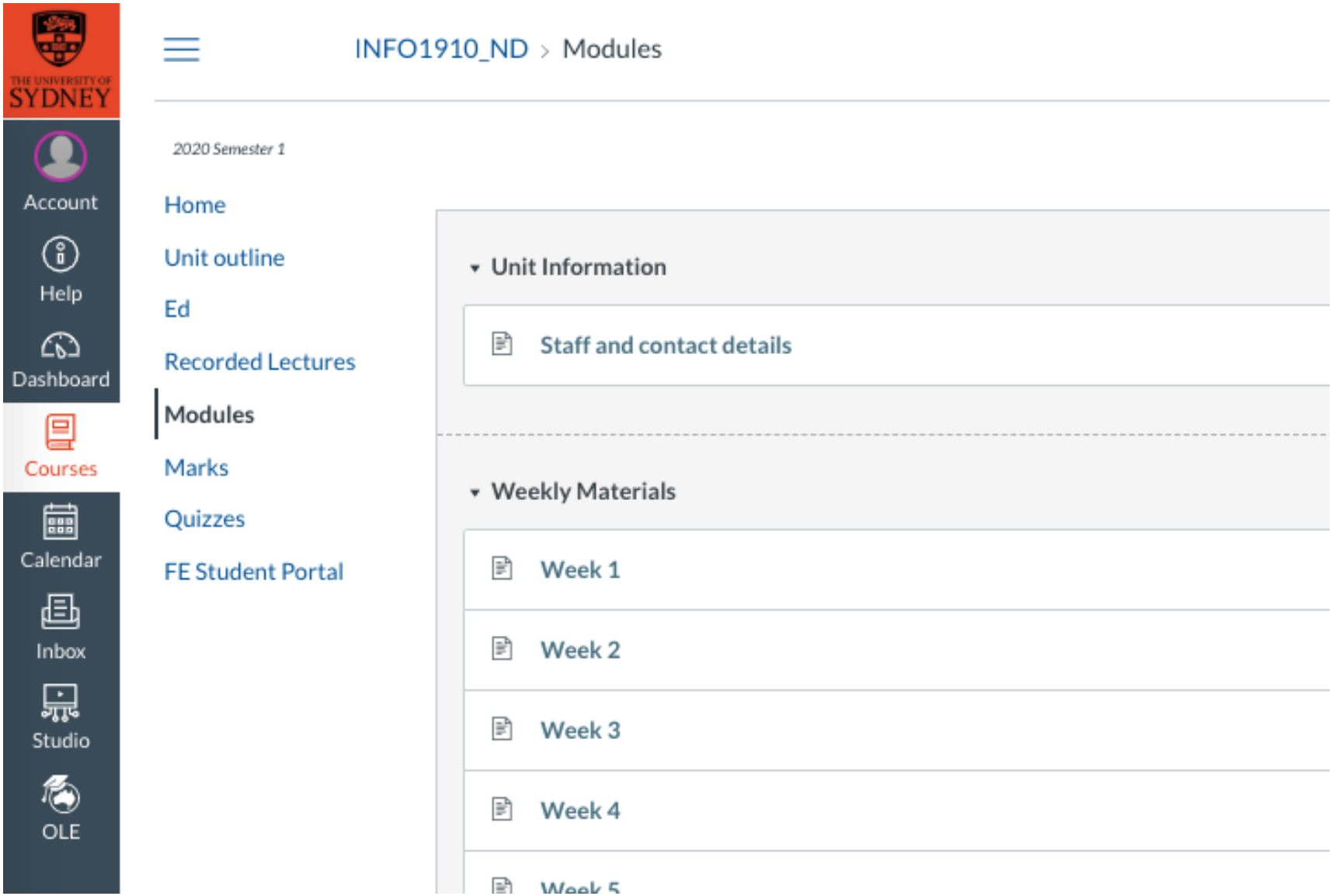
- Accessing your progressive grade
- Access to Live Lecture recordings
- Access to Lecture slides
- Access to Tutorial sheets
- Web links to other important places

canvas.sydney.edu.au

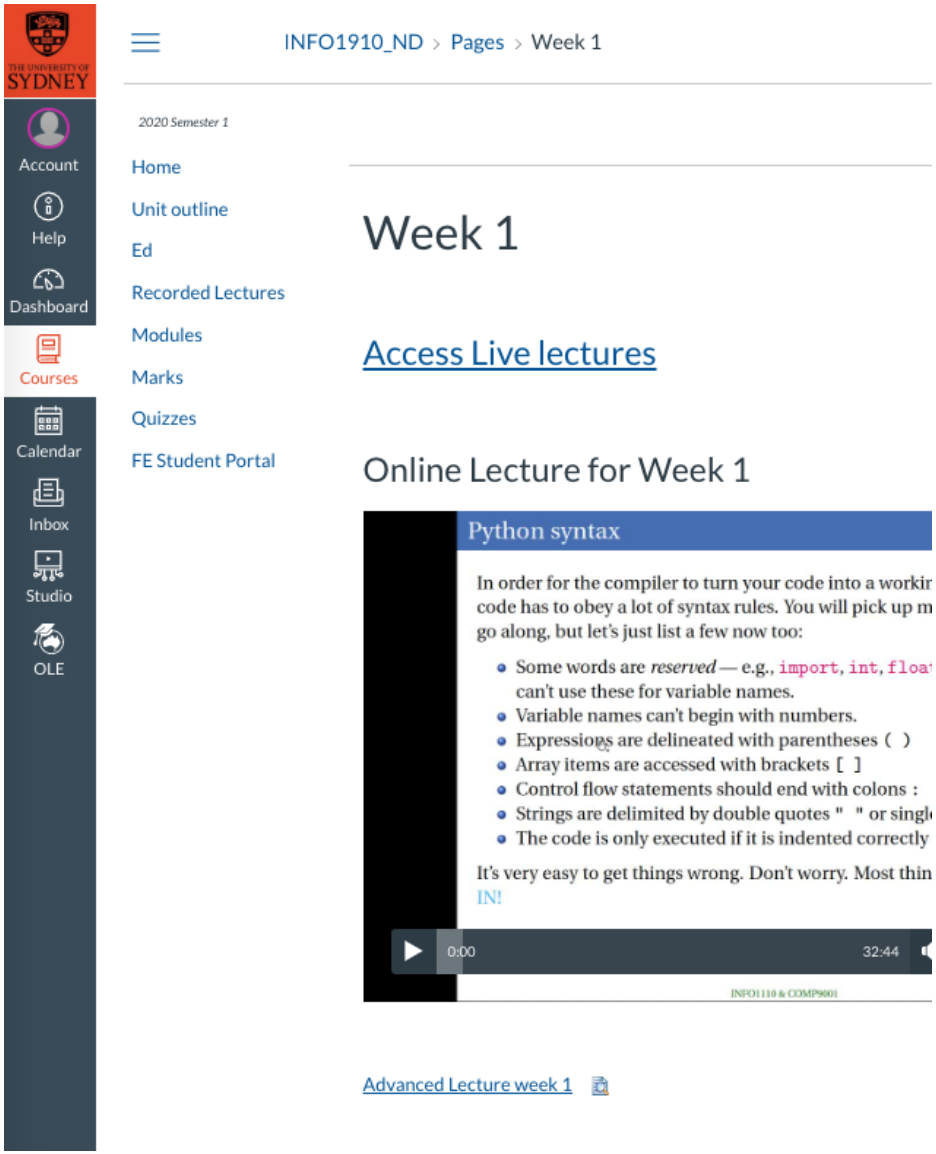
# Weekly modules via Canvas



# Weekly modules via Canvas (cont.)



# Weekly modules via Canvas (cont.)



# Where to get help?

1. Student admin → `https://sydneystudent.sydney.edu.au` or contact the Student Centre. Please check these first. E.g. timetable, passwords, payments, enrolments etc.
2. Ed discussion forum using the *1910* tag → Ed
3. Your tutor in your designated laboratory
4. Contact a teaching staff → Ed (private thread)
5. Contact Lecturer/Coordinator Dr. John Stavrakakis
   Please be patient in the case that I am delayed from the previous timeslot. Notice: all meetings are recorded.

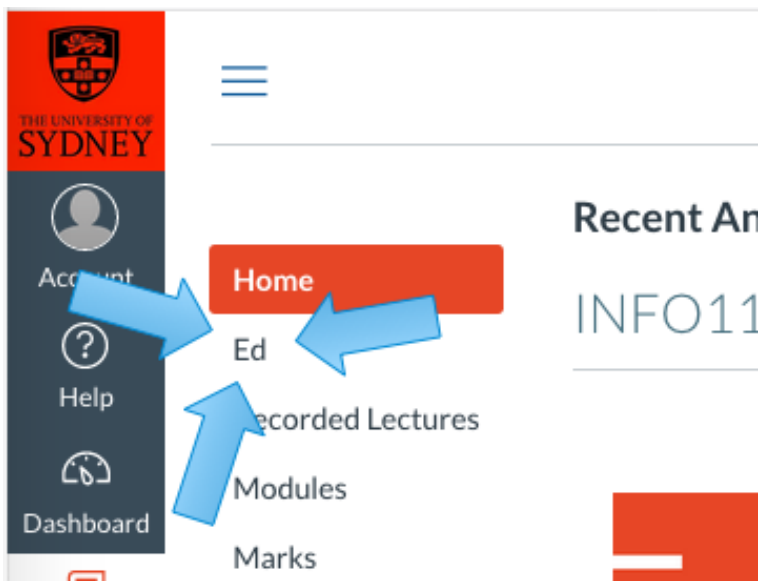
# Online resources: Ed

We use Ed for discussion and assessmentsEd

Assessment specification and submission portals are here. They will utilise git

**Submit online assessments here!**

Ed



# Assessments: INFO1910

| **Assessment** | **Due Date** | **Weighting** |
|---|---|---|
| Online problem | Weeks 2, 4, 7, 8, 10 | 10% |
| Tutorial quiz | Weeks 3, 5, 6, 9, 11 | 10% |
| Assignment 1 | 11:59 PM, 17 September (Week 7) | 15% |
| Assignment 2 | 11:59 PM, 05 November (Week 13) | 20% |
| Final Exam | Exam Period | 40% |

# Assessments: INFO1910 description

Online problem - 5 programming problems due in certain weeks of semester. To be completed in your own time. This can cover material for standard, advanced, or both. Best 4/5.

Lab task - In 5 weeks of the semester, there will be a lab activity. This can evaluate your knowledge in contents for standard, advanced, or both. Failure to attend results in zero. Best 4/5.

# Assessments: INFO1910 description (cont.)

Assignments - strictly for advanced students. Available in Ed, specification and submission. Discussions will take place in their own threads using the '1910' category. Additional functionality or requirements will be made for advanced compared with standard. This may require using a different programming language for a part or all of the assignment.

Final examination - This can evaluate your knowledge in contents for standard, advanced, or both. Harder questions on standard topics, questions on advanced only concepts, and will require writing code in different programming languages. Computer exam + Oral exam.

# Attendance and other assessments

We still encourage advanced students to attempt and complete the INFO1110 weekly task as a way to confirm your own understanding of the subject matter.

INFO1110 seminar attendance is optional, but always welcomed. Online recording is available.

Lecture attendance is necessary, however, a lecture recording is available online. This may be helpful if you wish to fast forward or skip sections as the pace may understandably be slow.

You must attend your lab. See your timetable for the lab details.

# Warning: This course is harder

The standard version of this course, INFO1110 is considered difficult by students. There is a high failure rate.

This Advanced course is a level much harder than that. Students also fail this course.

There are students from INFO1910 that fail this course, whereas they could pass INFO1110. It is however still a fail outcome.

The entrance examination is more of a self evaluation. It should help you reflect on whether advanced is right for you.

If you are finding that the material for INFO1910 is too much, you should change enrolment to INFO1110 before week 3.

# Get help when you need it!

Even though you are enrolled in advanced, it does not mean you cannot ask simple questions. You are still learning after all! please use Ed where you can and formulate your question to attract answers from anyone who can.

Advanced students can also attend the helpdesk for all students throughout the semester. Though this support may be limited to help standard, many of the tutors are technically competent, but not for all programming languages. You may find value in being part of the session even as an eavesdropper!

# Let's get into UNIX!

Why, where and how

Creating files

Redirection and piping

Version control system: git

man pages

Compiling a python or C program

Basic Bash programming