
COSC1229/1479 Computational Science 1 Course Overview

Where are we ?

- COSC1229/1479: Computational Science 1
- Course Leader/Lecturer: Geoff Leach - <http://www.cs.rmit.edu.au/~gl>
 - gl@cs.rmit.edu.au, Office: 10.9.18, Phone: x53207
- Lecturer: Ron van Schyndel - <http://www.cs.rmit.edu.au/~ronvs>
 - ronvs@cs.rmit.edu.au, Office: 14.6.4, Phone: x59677
 - Consultation time: Friday 3.00 - 5.00pm (or after lecture/tute)
- Tutor: Andrew Charles
 - andrew.charles@rmit.edu.au
- Abbreviation: Computational Science = CLS
- News:
 - Group is `rmit.cs.ComputationalScience1`
 - news://news.cs.rmit.edu.au/rmit.cs.ComputationalScience1
 - or
 - <https://inside.cs.rmit.edu.au/webnews/> then subscribe to `rmit.cs.ComputationalScience1`

Who are You ?

- There are a variety of students currently enrolled in this course:
 - Undergraduate:
 - B. App. Sc. (Computational Science), 16 students
 - B. App. Sc. (Computer Science), 9 students
 - B. App. Sc. (Information Technology) 4 students
 - Post-Graduate:
 - Grad. Dip. Software Development 2 students
 - M Tech (IWEB) 1 student

Computational Science 1

- Part of the Graphics group of courses
 - COSC1212/1492 - Graphics Programming using Java3D
 - INTE1113/2033 - 3D Web Technologies
 - COSC1187/1186 - Interactive 3D Graphics & Animation
 - COSC1224/1226 - Real-Time Rendering & 3D Games Prog.
 - COSC1213/1215 - Web Media Technologies
 - COSC2271/2272 - Digital Media Programming

Course Overview

- 12 Lectures, 12 tutorials
- 2 Assignments (30% each),
 - Handed out in weeks 3 and 8, submitted by end of week 7 and 12
 - Both assignments to be demonstrated in the following week (8 and 13)
 - Students to set up their account
<http://yallara.cs.rmit.edu.au/~yourid/CLS/index.html>
 - Assignments to be submitted using turnin
- A 3 hour Exam (40%)
- Hurdles:
 - Successful completion in the combined assignment marks
 - Successful completion in the exam



Syllabus

1. <no lecture>
2. Intro and Overview,
3. Fundamentals of Numerical Analysis
4. Kinematics, Collision Dynamics
5. Computational Complexity
6. Physically-based Modelling, Particle Collisions, Pendulum motion
7. Spring-Mass Simulations, Projectiles, Multiple Spring Systems
8. Overview of Matlab and Octave
9. Some Elements of 1D Signal Processing
10. Introduction to Chaos
11. Fractal Geometry and Chaos
12. Monte-Carlo Methods
13. Revision



Outcomes

The experience we hope to gain from this course is:

- to understand the foundations of computational science;
- to recognise key problems in computational science;
- to understand and apply fundamental numerical methods;
- to understand the role of visualisation in computational science and apply some of its techniques;
- to implement simulation programs.



Assignments

Two assignments will constitute 30% each of your final mark.

- They will consist of Java based application or applet and supporting documentation.
- Students should work individually on the assignment, but discussion between students, particularly during tutes, is encouraged. Take advantage of tutes to bring up problems and ideas as though the assignment was a project in a commercial environment.



Do NOT share code!!! Two people working on an idea together, one writing the code, then sharing that code is **not acceptable!**

Plagiarism is an Offence

Plagiarism:

- * Submitting an assignment that contains other people's work.
- * Helping other students to plagiarise

All submitted work must be your own.

The only exception: Other people's work can be included if the assignment has explicit instructions to do so. All copied work (from the internet, other students, or staff) **must** be clearly identified.

A student who submits copied work without reference receives **no marks** for that assignment. Partial marks will not be given. Even if only part of the assignment was copied, the student will get **no marks**. If this means that a hurdle is not reached, then the student **fails** the subject.

A student who plagiarises a second time will be sent to the university disciplinary committee. Penalties include expulsion from the university.

For more information, see "plagiarism" under
<http://www.cs.rmit.edu.au/students>

University Plagiarism Statement

Students are reminded that cheating, whether by fabrication, falsification of data, or plagiarism, is an offence subject to University disciplinary procedures. Plagiarism in oral, written or visual presentations is the presentation of the work, idea or creation of another person, without appropriate referencing, as though it is one's own. Plagiarism is not acceptable. The use of another person's work or ideas must be acknowledged. Failure to do so may result in charges of academic misconduct which carry a range of penalties including cancellation of results and exclusion from your course.

Students are responsible for ensuring that their work is kept in a secure place. It is also a disciplinary offence for students to allow their work to be plagiarised by another student. Students should be aware of their rights and responsibilities regarding the use of copyright material.

Exam

The exam will be three hours and will make up 40% of the final mark.

It will not involve writing complete programs, but may require modification or discussion of existing code samples, correction of erroneous code, or writing pseudo-code..



Help Desks

- Java / C Help Desk

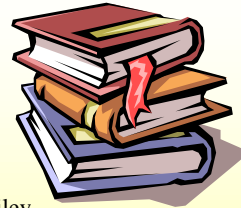
- Assistance with Java programming
- Timetable for help desk attendance can be found at <http://www.cs.rmit.edu.au/timetables/city/2002s2> under *CS Help Desk – Java or C*
- Currently Java help is on Tue/Thu/Fri 16.30-19.30
Currently C help is on Tue/Wed/Fri 16.30-19.30

Recommended Texts

Class notes will be available in print form prior to each lecture and parts may be online before that.

Prescribed References:

- David Acheson, *From Calculus to Chaos, An introduction to Dynamics*. Oxford University Press, 1997

**Strongly Recommended:**

- R. Landau, M Paez, *Computational Physics*, 1997, Wiley

Recommended References:

- David Halliday, Robert Resnick and Jearl Walker, *Fundamentals of Physics*, Sixth Edition (parts 1 and 2), 1999
- William H. Press et. al., *Numerical Recipes*, (in whichever programming language the student is familiar)

Finally

Have Fun !

