UNSW Course Outline

ZPEM1307 Computational Problem Solving - 2024

Published on the 20 Feb 2024

General Course Information

Course Code : ZPEM1307
Year : 2024
Term : Semester 1
Teaching Period : Z1
Is a multi-term course? : No
Faculty : UNSW Canberra
Academic Unit : UC Science
Delivery Mode : In Person
Delivery Format : Standard
Delivery Location : UNSW Canberra at ADFA
Campus : UNSW Canberra
Study Level : Undergraduate
Units of Credit : 6

Useful Links

Handbook Class Timetable

Course Details & Outcomes

Course Description

Computation underpins our 'daily digital life': Students taking this course will receive a foundational background in computational problem-solving. On the theoretical side this includes an introduction to logic, along with the scientific and engineering methodologies, design

principles, and a grounding in the theories and models of computation. In parallel practical skills of problem structuring, solution design, algorithm writing, structured programming, and data representation within a computational environment will be acquired. Theoretical and practical lessons are contextualised in the modern IT environment with introductions to core computer science topics such as operating systems, networks, simulation, and programming languages. Students will design and implement a number of solutions to computational problems - bringing together the theory, milieu, and practical skills acquired in the course.

Course Aims

Course Learning Outcomes

Course Learning Outcomes
CLO1 : Demonstrate an understanding of how to computationally approach and solve structured problems.
CLO2 : Apply a software engineering process of design-build-test to realise solutions to small computational problems.
CLO3 : Employ the control-structures and data structures of a modern programming language to implement solutions to small computational problems.
CLO4 : Demonstrate the ability to adequately solve small problems from the domains of engineering, physical and computational science, and cybersecurity.

Course Learning Outcomes	Assessment Item
CLO1 : Demonstrate an understanding of how to computationally approach and solve structured problems.	• Competency-based Labs • Class Test • Exam
CLO2 : Apply a software engineering process of design-build-test to realise solutions to small computational problems.	• Competency-based Labs
CLO3 : Employ the control-structures and data structures of a modern programming language to implement solutions to small computational problems.	• Class Test • Exam • Competency-based Labs
CLO4 : Demonstrate the ability to adequately solve small problems from the domains of engineering, physical and computational science, and cybersecurity.	• Competency-based Labs

Learning and Teaching Technologies

Moodle - Learning Management System | Echo 360

Assessments

Assessment Structure

Assessment Item	Weight	Relevant Dates
Competency-based Labs Assessment Format: Individual	60%	Due Date: Week 4: 18 March - 22 March, Week 8: 29 April - 03 May, Week 10: 13 May - 17 May, Week 13: 03 June - 07 June
Class Test Assessment Format: Individual	10%	Due Date: Week 6: 01 April - 05 April
Exam Assessment Format: Individual	30%	Due Date: Exam period

Assessment Details

Competency-based Labs

Assessment Overview

Weekly lab sessions, with 4 submissions throughout the semester (15% weighting each) comprising written problem breakdown and algorithm, code with comments indicating thought process and methodology, and verbal reflection via presentation in the lab sessions. Practical component of the assessment showing ability to produce working code and assess its effectiveness at solving the problem, with less emphasis on computational elegance or efficiency.

Course Learning Outcomes

- CLO1 : Demonstrate an understanding of how to computationally approach and solve structured problems.
- CLO2 : Apply a software engineering process of design-build-test to realise solutions to small computational problems.
- CLO3 : Employ the control-structures and data structures of a modern programming language to implement solutions to small computational problems.
- CLO4 : Demonstrate the ability to adequately solve small problems from the domains of engineering, physical and computational science, and cybersecurity.

Class Test

Assessment Overview

Theory-based assessment of higher level understanding of programming and problem solving principles. Scaffolds the exam.

Course Learning Outcomes

- CLO1 : Demonstrate an understanding of how to computationally approach and solve

structured problems.
- CLO3 : Employ the control-structures and data structures of a modern programming language to implement solutions to small computational problems.

Assessment Length

1 hour

Exam

Assessment Overview

Theory-based assessment of higher level understanding of programming and problem solving principles.

Course Learning Outcomes

- CLO1 : Demonstrate an understanding of how to computationally approach and solve structured problems.
- CLO3 : Employ the control-structures and data structures of a modern programming language to implement solutions to small computational problems.

Assessment Length

2 hours

General Assessment Information

The course assessement consists of the lab component (60% weighting), a class test (10% weighting) and an exam (30% weighting). The lab component is mandatory, while the class test and exam are optional.

Each of the four lab submssions will be assessed to determine whether the learning outcomes of the lab have been achieved, with successful completion of each receiving 15% weighting. Sucessful completion of all four lab submissions is sufficient to pass the course, with a mark of 60% being awarded.

The class test and exam may be undertaken to increase the mark beyond 60%, but are not required to pass.

Use of Generative Artificial Intelligence (AI) such as ChatGPT in lab submissions

You can use generative AI software in lab submissions to the extent specified in the assessment instructions and as advised by lecturers in class. Any output of generative software within your assessment must be attributed with full referencing.
If the outputs of generative AI such as ChatGPT form part of your submission and is not

appropriately attributed, it will be regarded as serious academic misconduct and subject to the standard penalties, which may include 00FL, suspension and exclusion.
* To cite: OpenAI (Year Accessed). ChatGPT. OpenAI. https://openai.com/models/chatgpt/
* Please note that the outputs from these tools are not always accurate, appropriate, nor properly referenced. You should ensure that you have moderated and critically evaluated the outputs from generative AI tools such as ChatGPT before submission.

Grading Basis

Standard

Requirements to pass course

Grade of 50% or greater.

Course Schedule

Teaching Week/Module	Activity Type	Content
Week 1 : 26 February - 1 March	Lecture	Problem solving and algorithm development. Introduction to the Python programming environment.
	Tutorial	Further exploration of lecture content.
Week 2 : 4 March - 8 March	Lecture	Problem solving and algorithm development. Python intro – variables, assignment, vectors and arrays, strings, maths operations, built-in functions, graphics, libraries.
	Tutorial	Further exploration of lecture content.
	Laboratory	Lab topic 1.
Week 3 : 11 March - 15 March	Lecture	Problem solving and algorithm development. Scripts and functions. Generative AI as a programming tool.
	Tutorial	Further exploration of lecture content.
	Laboratory	Lab topic 1.
Week 4 : 18 March - 22 March	Lecture	Input/output. Decision/control statements, loops. Numerical errors, uncertainty, bugs. Testing, evaluation, validation.
	Tutorial	Further exploration of lecture content.
	Assessment	Submission of Lab topic 1.
Week 5 : 25 March - 29 March	Lecture	Testing, evaluation, validation. Error handling.
	Tutorial	Further exploration of lecture content.
	Laboratory	Lab topic 2.
Week 6 : 1 April - 5 April	Lecture	Analysis concepts: Curve fitting, regression, interpolation, graphics.
	Tutorial	Further exploration of lecture content.
	Laboratory	Lab topic 2.
	Assessment	Class test (optional).
Week 7 : 22 April - 26 April	Lecture	Analysis concepts: Curve fitting, regression, interpolation, graphics.
	Tutorial	Further exploration of lecture content.
	Laboratory	Lab topic 2.
Week 8 : 29 April - 3 May	Lecture	Analysis concepts: Optimisation, search, maximisation/minimisation, roots.
	Tutorial	Further exploration of lecture content.
	Assessment	Submission of Lab topic 2.
Week 9 : 6 May - 10 May	Lecture	Analysis concepts: Linear systems, matrix algebra and inversion.
	Tutorial	Further exploration of lecture content.
	Laboratory	Lab topic 3.
Week 10 : 13 May - 17 May	Lecture	Analysis concepts: ordinary differential equations.
	Tutorial	Further exploration of lecture content.
	Assessment	Submission of Lab topic 3.
Week 11 : 20 May - 24 May	Lecture	Analysis concepts: ordinary differential equations.
	Tutorial	Further exploration of lecture content.
	Laboratory	Lab topic 4.
Week 12 : 27 May - 31 May	Lecture	Introduction to Matlab.
	Tutorial	Further exploration of lecture content.
	Laboratory	Lab topic 4.
Week 13 : 3 June - 7 June	Lecture	Revision.
	Tutorial	Further exploration of lecture content.
	Assessment	Submission of Lab topic 4.

Attendance Requirements

Students are strongly encouraged to attend all classes and review lecture recordings.

Course Resources

Prescribed Resources

There are three resources which will be used in the course.

- *Applied Numerical Methods with Python for Engineers and Scientists*, Steven Chapra and David Clough (2021). McGraw-Hill Higher Education (International). https://bookshelf.vitalsource.com/books/9781265009502.
- *Think Python, How to Think Like a Computer Scientist*, Allen Downey (2015). Green Tea Press, available for free download from https://greenteapress.com/wp/think-python-2e/.
- *Learn Python With Jupyter*, Serena Bonaretti, available for free download from https://www.learnpythonwithjupyter.com.

Course Evaluation and Development

The course assessment has been re-designed for 2024 following feedback from previous years. Many students in Year 1 have no prior formal programming experience, and have found learning a programming language daunting. A compentency-based approach to achieve a passing grade has been adopted, to allow students to learn and explore new concepts, and develop and demonstrate practical skills in coding and problem solving for the types of problems they are likely to encounter in their later degree courses.

Staff Details

Position	Name	Email	Location	Phone	Availability	Equitable Learning Services Contact	Primary Contact
Convenor	John Young		Bldg 20 Rm 134	(02) 5114 5228	09:00 - 16:00 Monday to Friday (email to schedule a meeting if not in office).	No	Yes
Lecturer	Tapabrata Ray		Bldg 17 Rm 202		09:00 - 17:00 Monday to Friday	No	No
Demonstrator	Ali Ahrari		Bldg 15 Rm 113		09:00 - 17:00 Monday to Friday	No	No
Lecturer	Mo Hossny			(02) 5114 5363		No	No
Demonstrator	Parth Girdhar					No	No
	Heba El-Fiqi		Bldg 15 Rm 204	(02) 5114 5332	Office Hours Tuesdays 13:00 - 14:00	No	No

Other Useful Information

Academic Information

Course Evaluation and Development

One of the key priorities in the 2025 Strategy for UNSW is a drive for academic excellence in education. One of the ways of determining how well UNSW is progressing towards this goal is by listening to our own students. Students will be asked to complete the myExperience survey towards the end of each course.

Students can also provide feedback during the semester via: direct contact with the lecturer, the "On-going Student Feedback" link in Moodle, Student-Staff Liaison Committee meetings in schools, informal feedback conducted by staff, and focus groups (where applicable). Student opinions really do make a difference. Refer to the Moodle site for your course to see how the feedback from previous students has contributed to the course development.

Important note: Students are reminded that any feedback provided should be constructive and professional and that they are bound by the Student Code of Conduct.

https://www.gs.unsw.edu.au/policy/documents/studentcodepolicy.pdf

Equitable Learning Services (ELS)

Students living with neurodivergent, physical and/or mental health conditions or caring for someone with these conditions may be eligible for support through the Equitible Learning Services team. Equitable Learning Services is a free and confidential service that provides practical support to ensure your mental or physical health conditions do not adversely affect your studies.

Our team of dedicated **Equitable Learning Facilitators** (ELFs) are here to assist you through this process. We offer a number of services to make your education at UNSW easier and more equitable.

Further information about ELS for currently enrolled students can be found at: https://www.student.unsw.edu.au/equitable-learning

Academic Honesty and Plagarism

UNSW has an ongoing commitment to fostering a culture of learning informed by academic integrity. All UNSW staff and students have a responsibility to adhere to this principle of academic integrity. All students are expected to adhere to UNSW's Student Code of Conduct. Find relevant information at: Student Code of Conduct (unsw.edu.au)

Plagiarism undermines academic integrity and is not tolerated at UNSW. It is defined as using the words or ideas of others and passing them off as your own, and can take many forms, from deliberate cheating to accidental copying from a source without acknowledgement.

For more information, please refer to the following:

https://student.unsw.edu.au/plagiarism

Submission of Assessment Tasks

Special Consideration

Special Consideration is the process for assessing and addressing the impact on students of short-term events, that are beyond the control of the student, and that affect performance in a specific assessment task or tasks.

Applications for Special Consideration will be accepted in the following circumstances only:

- Where academic work has been hampered to a substantial degree by illness or other cause;
- The circumstances are unexpected and beyond the student's control;
- The circumstances could not have reasonably been anticipated, avoided or guarded against by the student; and either:

(i) they occurred during a critical study period and was 3 consecutive days or more duration, or a total of 5 days within the critical study period; or

(ii) they prevented the ability to complete, attend or submit an assessment task for a specific date (e.g. final exam, in class test/quiz, in class presentation)

Applications for Special Consideration must be made as soon as practicable after the problem occurs and at the latest within three working days of the assessment or the period covered by the supporting documentation.

By sitting or submitting the assessment task the student is declaring that they are fit to do so and cannot later apply for Special Consideration (UNSW 'fit to sit or submit' requirement).

Sitting, accessing or submitting an assessment task on the scheduled assessment date, after applying for special consideration, renders the special consideration application void.

Find more information about special consideration at: https://www.student.unsw.edu.au/special/consideration/guide

Or apply for special consideration through your MyUNSW portal.

Late Submission of assessment tasks (other than examinations)

UNSW has a standard late submission penalty of:

- 5% per day,
- capped at five days (120 hours) from the assessment deadline, after which a student cannot submit an assessment, and
- no permitted variation.

Students are expected to manage their time to meet deadlines and to request extensions as early as possible before the deadline.

Electronic submission of assessment

Except where the nature of an assessment task precludes its electronic submission, all assessments must be submitted to an electronic repository, approved by UNSW or the Faculty, for archiving and subsequent marking and analysis.

Release of final mark

All marks obtained for assessment items during the session are provisional. The final mark as published by the university following the assessment review group meeting is the only official mark.