



UNSW
College

DPST1092 / CPTG1392

**Computer Systems
Fundamentals**

Term 3 2026

**COURSE OUTLINE
DIPLOMA PROGRAM**

For detailed information about copyright, ownership and delivery of this course, see [page 2](#).

This course is part of the following programs:

Program Name	CRICOS Course Code
Diploma of Computer Science	113046B
Diploma of Engineering	113047A

Copyright Notice [Include this notice for courses with UNSW students enrolled]

The materials for this course, including but not limited to lectures, assignments, assessments, supplementary resources, and this Course Outline, are protected under the Copyright Act 1968 (the Act) and international intellectual property laws.

The course materials are owned by the University of New South Wales (UNSW) and are delivered by UNSW Global Pty Limited (trading as UNSW College) on behalf of UNSW Sydney. Each entity retains ownership of their respective contributions to the course.

By enrolling in this course, students are granted a non-exclusive, non-transferable license to use the course materials strictly for personal educational purposes. Any reproduction, distribution, adaptation, or commercial use of these materials, in whole or in part, without explicit written permission from UNSW or UNSW College, as applicable, is strictly prohibited. This includes, but is not limited to, sharing materials on external websites or platforms, photocopying, or redistributing them to third parties.

UNSW retains ownership of the course content and reserves the right to modify, update, or withdraw these materials at its discretion. Students are encouraged to respect the intellectual property rights of both entities and comply with the licensing terms provided.

This Course Outline has been provided to you by UNSW College for use in 2025. Any unauthorised reproduction or communication of this material may infringe copyright laws and result in legal consequences.

TABLE OF CONTENTS

TABLE OF CONTENTS	2
SECTION 1.....	4
Staff.....	4
Emailing Staff	4
What is expected of you.....	5
What you can expect from your teachers	5
Changes to this course as a result of student feedback	6
Policy and how it affects you.....	6
Academic Integrity	6
Artificial Intelligence	6
Integrated Work-based Learning [Applied Programs Only]	Error! Bookmark not defined.
SECTION 2.....	7
Course Information	7

UNSW College

Building L5, UNSW Sydney Campus, 223 Anzac Parade, Kensington NSW 2033 Australia

T: +61 (2) 8936 2222 | W: unswcollege.edu.au

UNSW Global Pty Limited ABN 62 086 418 582 trading as UNSW College™. UNSW College CRICOS Provider Code 01020K. UNSW College TEQSA Provider ID: PRV13020 (Institute of Higher Education)

Course Summary	8
Course Aims	8
Course Learning Outcomes	8
Program Learning Outcome Alignment	8
Strategies and Approaches to Learning and Teaching	9
Methodology	10
Course Schedule and Structure	12
SECTION 3	13
Assessment details	13
Assessment Schedule	13
Assessment Guidelines	14
Assessment 1 – Assessment Name	Error! Bookmark not defined.
Assessment 2 – Assessment Name	Error! Bookmark not defined.
Assessment 3 – Assessment Name	Error! Bookmark not defined.
SECTION 4	21
Essential Resources for Students	21
Support for Students	21
References	21

SECTION 1

This Course Outline is designed to guide you through the course successfully. It includes information about the course, your teaching team, how to prepare for classes, details of your assessment tasks, and where to access support and assistance when needed.

Staff

Position	Name	Email
Course Convenor	Sasa Maric	s.maric@unswcollege.edu.au
Admin/Tutor	Stephen Vinall	s.vinall@unswcollege.edu.au

Emailing Staff

When contacting staff, you must use your official UNSW College email address. In all emails, please include your:

- ZID
- First and Surname
- Course Name and Course Code

Your teachers are your first point of contact for academic advice or support with your coursework. Their email contact details are easily accessible on your online learning platform. If you have questions about your work, don't hesitate to reach out—they are available to assist you and will respond promptly.

As a secondary point of contact, students may use the official teams channel.

What is expected of you

ATTENDANCE

To achieve academic success, it is recommended that you aim to achieve 100% attendance. You are expected to attend at least 80% of classes. Educational research consistently demonstrates that this level of attendance is associated with a high likelihood of achieving a passing grade.

CHECKING YOUR STUDENT EMAIL

It is important to check your student email account regularly, as all official communications, updates, and announcements related to your course will be sent there. Staying up to date with your email will ensure you do not miss any critical information or deadlines.

ONLINE LEARNING

Access to the course's online learning platform is an essential part of your studies, as all course content will be available online. You are expected to regularly log in to the platform to access lectures, assignments, resources, and updates.

YOUR CONDUCT

You are required to:

- act honestly and uphold academic integrity in all your studies
- treat all students, staff, and affiliates with courtesy and respect
- attend scheduled learning activities and engage actively in your studies
- submit your own work for all assessment tasks and avoid misconduct such as plagiarism or cheating
- use College resources, facilities, and equipment responsibly
- keep your student ID and passwords secure and confidential
- follow all policies, procedures, and lawful directions from staff
- refrain from behaviour that disrupts the teaching and learning environment or negatively impacts others
- maintain the College's reputation through responsible and respectful conduct.

What you can expect from your teachers

TEACHING SUPPORT AND RESPONSIBILITIES

The teaching team is dedicated to supporting your learning and success in this subject. You can expect the teaching team to:

- be well-prepared for each teaching session
- clearly explain course requirements and content
- provide assistance at prearranged times to address your questions or concerns
- deliver constructive and timely feedback on assessments.

PROFESSIONAL CONDUCT

The teaching team is also expected to:

- promptly report any concerns regarding academic or non-academic misconduct
- respond to student queries and concerns about the subject or relevant policies and procedures
- treat all students and staff fairly, respectfully, and in accordance with all policies and procedures.

Changes to this course as a result of student feedback

Student feedback plays a key role in improving this course and we encourage you to participate in any opportunity that allows you to provide feedback, including the My Experience surveys at the end of your teaching session. Changes may be implemented progressively to enhance your learning experience (and those of future students), such as adjustments to teaching materials, assessment tasks, or learning activities. Your input helps ensure the course remains relevant, engaging, and supportive of your academic success. Please note that all surveys are conducted anonymously and, as such, your name or student ID cannot be identified.

Policy and how it affects you

There are multiple policies that apply to students, and it is important to ensure you are referring to the correct policies for the program you are enrolled in. The relevant policies for your program, depending on your enrolment date and program type, are highlighted on the UNSW College website at: <https://www.unswcollege.edu.au/about/policies/policy-documents#policies>. Please review this page carefully to ensure you are accessing the appropriate policy documents, including:

- Academic Integrity Policy
- Assessment Policy
- Attendance Monitoring Policy
- Enrolment Policy
- IT Security Policy
- Sexual Misconduct Policy
- Student Health, Safety and Wellbeing Policy
- Student Grievances and Complaints Policy
- Student Misconduct Policy

Academic Integrity

Academic integrity requires you to act with honesty, trust, fairness, respect, responsibility, and courage. It is your responsibility to uphold these values by ensuring your work is your own and by appropriately acknowledging the contributions and ideas of others.

Please adhere to the UNSW College Academic Integrity Policy and Procedure (<https://www.unswcollege.edu.au/about/policies/policy-documents#policies>) when completing and submitting your assessments.

You are responsible for reading, understanding, and following the instructions about:

- Detecting breaches of academic integrity
- Procedural Fairness
- Investigating a breach of academic integrity
- Outcome of an investigation
- Levels of breach and penalties
- Communications and notifications about academic misconduct
- Recording breaches of academic misconduct
- Appeals

Artificial Intelligence

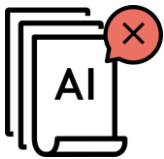
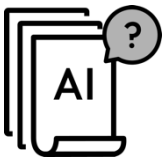
UNSW College has created its own Artificial Intelligence Guidelines called the 3Cs Model. You are required to adhere to these guidelines when you complete your assessments. These guidelines show you when it is appropriate to use artificial intelligence for your assignment and when it is not permitted.

UNSW College

Building L5, UNSW Sydney Campus, 223 Anzac Parade, Kensington NSW 2033 Australia

T: +61 (2) 8936 2222 | W: [unswcollege.edu.au](https://www.unswcollege.edu.au)

UNSW Global Pty Limited ABN 62 086 418 582 trading as UNSW College™. UNSW College CRICOS Provider Code 01020K. UNSW College TEQSA Provider ID: PRV13020 (Institute of Higher Education)

Category 1: AI tools cannot be used	Category 2: AI tools can be used in assistive role	Category 3: AI tools can be used for integration
 For this category, you need to show basic skills like remembering, understanding, and using your knowledge, which are essential for future studies and work. For example, you might need to remember and use a formula to solve a math question during a final exam or explain why a business solution is important to your colleagues. In this category, you cannot use AI tools at all. This includes in-person exams, class tests, oral exams, some labs and practicals, and discussion-based assessments. Your teacher will further explain why this category is important.	 For this assessment, you may use AI tools to help you develop specific skills. For example, you might use AI to look at data, find patterns in text, and get new ideas about a topic or question. You may use AI tools to help with certain tasks in the assessment. For example, you might use AI to draft and organise content, prepare for exams, test code, translate content, give feedback, and proofread. Your teacher will explain why this category is used to make sure everyone has a fair experience. Some parts of the assessment will not allow AI.	 For this assessment, you may use AI tools at different stages of your assessment. In this category, you should show your skills in using AI tools to solve problems, make decisions, and create solutions. This assessment may include a part where you must show that you can use AI in an ethical and responsible way. For example, you might use AI to generate ideas, compare content, produce summaries, analyse content, reframe content, research and find answers, and write content that other students can review and give you feedback on. Your teacher will support and guide you to ensure everyone has a fair experience.

For each assessment in your course guideline, you will see one of the category icons (Category 1, 2 or 3). You can read about the category for your assessment under the heading **Supporting Information**.

If you choose to use artificial intelligence for an assessment that prohibits the use of AI tools, you will be penalised for academic misconduct.

SECTION 2

Course Information

Course code	DPST1092/CPTG1392
Course name	Computer Systems Fundamentals
Units of credit	6
Course level	(Diploma)/Level 2
Assumed knowledge	DPST1091/CPTG1391

Course Summary

This course provides a programmer's view on how a computer system executes programs and manipulates data. It enables students to become effective programmers in dealing with issues of performance, portability, and robustness. The course assumes that students have completed the first course in programming in the C programming language, DPST1091/CPTG1392. **Course Aims**

Course Learning Outcomes

This course aims to give students an overview of the structure and behaviour of modern computer systems, and to provide a foundation for later courses on operating systems, computer architecture and compilers, where a deeper understanding of systems-level issues is required.

On successful completion of this course with a passing grade, you will be able to:

1. Describe the layers of architectures in modern computer systems from hardware device levels upwards
2. Describe the principles of memory management and explain the workings of a system with virtual memory management
3. Explain how the major components of a CPU work together, including how data (including instructions) are represented in a computer
4. Design, implement and analyse small programs at the assembly/machine level
5. Describe the relationship between high-level procedural languages (e.g., C) and assembly/machine language in the conventional machine layer, including how a compiled program is executed in a classical von Neumann machine
6. Explain how input/output operations are implemented, and describe some basic I/O devices
7. Describe the components comprising and the services offered by an operating system
8. Implement simple programs involving inter-process communication and concurrency

Program Learning Outcome Alignment

Course Learning Outcome (CLO)	Program Learning Outcome (PLO)	Related Assessment
CLO1:	Understanding of underpinnings (EA1.1)	Exam
CLO2:	Understanding of specialist bodies of engineering knowledge (EA1.3)	Exams and labs
CLO3:	Understanding of specialist bodies of engineering knowledge (EA1.3)	Exam, Tests, Labs
CLO4:	Application of established engineering practice (EA2.1)	Labs, Tests, Assignments
CLO5:	Conceptual understanding of computer underpinnings (EA1.2)	Exam, Tests, Labs
CLO6:	Understanding of specialist bodies of engineering knowledge (EA1.3)	Exam, Labs
CLO7:	Understanding of specialist bodies of engineering knowledge (EA1.3)	Exam, Labs
CLO8:	Application of established engineering practice (EA2.1)	Exam

Strategies and Approaches to Learning and Teaching

ATTENDANCE AND ENGAGEMENT REQUIREMENTS

This course will consist of x hours of classes each week [provide breakdown of the types of classes/practicals/workshops]. This should be supplemented by x hours of self-directed learning, which may include interacting with course content, online activities, research, and work on assessment tasks.

Face-to-face hours	Self-directed hours	Consultation hours
4 hours of Lectures		
2 hours of Tutorial	16	2 consultation hours
2 hours of Labs		

Lectures - 4 hours per week (weeks 1-6 and 8-12)

This course involves a number of teaching activities: Lectures present theory and concepts, by way of case studies and practical examples. Lecture notes will be provided in advance of each class. There will be 4 hours of timetabled lectures each week. Tutorials allow students to collaboratively work through example problems to illustrate lecture ideas, and have concepts from lectures clarified by the tutor. Each student has an opportunity to lead a "code review" (discussion of a piece of software or a system) at least once during the course.

Tutorials - 2 hours per week (weeks 1-6 and 8-12)

Lab Classes involve small exercises where students build systems that illustrate the ideas covered in lectures. To obtain a mark for a lab exercise you should submit it using give. If you don't finish it during the lab, you may continue working on it during the week, you must submit it (using give) by the following Monday 11:59pm in order to get any marks for it. You cannot obtain marks by e-mailing lab work to tutors.

Labs - 2 hours per week (weeks 1-6 and 8-12)

Lab Classes involve small exercises where students build systems that illustrate the ideas covered in lectures. To obtain a mark for a lab exercise you should submit it using give. If you don't finish it during the lab, you may continue working on it during the week, you must submit it (using give) by the following Monday 11:59pm in order to get any marks for it.

You cannot obtain marks by e-mailing lab work to tutors.

During the lab, your tutor will give you guidance and can provide feedback on your approach to the problem and on the style of your solution. Some labs may contain exercises which will be assessed during the lab.

The lab exercises for each week are worth in total 2 marks. We will sum to best 9 lab marks to give you a mark out of 18; if their sum exceeds 15, your total mark will be capped at 15 Most labs include one or more challenge exercises. Challenge exercises may involve concepts not covered in lectures and they range in difficulty from not-very-hard to almost-impossible.

The contribution of challenge exercises to lab marks will be limited to 20%; hence you can obtain nearly all (over 95%) marks available for the lab component without completing challenge exercises. If you wish to obtain a high mark for DPST1092, attempting some challenge exercises is highly recommended. If your goal is just to master the core material and pass DPST1092, you can ignore challenge exercises.

Assignments – 2 per term

Assignments are take-home problems that are larger in scope than Lab exercises and require students to use creativity to solve a challenging realistic problem. Each assignment requires students to understand the problem, design a solution, and implement and test their solution.

Online Forum

An online forum on teams allows students to ask and answer questions on the tutorial, lab and assignment exercises, and on lecture material.

Final Exam

There will be a final exam during the exam period. You will be required to attend the exam Face to Face.

It may contain a mixture of: implementation tasks (which will require you to write C and assembler programs); "theory" questions (which require analysis and written answers); multiple choice questions. During this exam you will be able to execute, debug and test your answers. The implementation tasks will be similar to those encountered in lab exercises.

There is a hurdle requirement on the final exam. If you do not score at least 40% on the exam, you cannot pass this course. If your overall course score exceeds 50%, despite scoring very poorly (<40%) on the exam, the hurdle will be enforced via a grade of UF. Of course, if your overall course score is less than 50%, then your grade will be FL.

Methodology

The methodology used in *Computer Systems Fundamentals (DPST1092/CPTG1392)* is designed to support a progressive and student-centred approach to learning complex systems-level programming and computer architecture concepts. This course adopts a blended methodology that includes lectures, tutorials, labs, assignments, and peer collaboration—each playing a critical role in guiding students from basic understanding to advanced problem-solving and practical application.

One of the central strategies used is **scaffolding**, which involves breaking down complex topics (like assembly programming, memory management, or concurrency) into manageable chunks and progressively building students' understanding. Early weeks start with foundational content (e.g., bit manipulation and C revision), which are then scaffolded into deeper systems concepts (e.g., file systems, processes, virtual memory) through carefully structured lab exercises and assignments. Each task is designed to extend the knowledge from the previous, allowing students to build confidence and competence incrementally.

Guided practice is another critical component. In tutorials and labs, students are given problems to work on collaboratively with support from tutors, encouraging active learning. Tutors provide immediate feedback, helping correct misconceptions and deepen understanding. Lab exercises often start with simpler tasks before introducing challenge exercises, reinforcing the scaffolding approach by allowing students to attempt more difficult problems once they have mastered the core material.

Constructive alignment is evident throughout the course. Learning activities and assessments are directly aligned with the Course Learning Outcomes (CLOs). For instance, writing assembly programs directly supports CLO 4, while manipulating file metadata in C targets CLOs 2 and 8. This alignment ensures that all assessments are meaningful and provide students with opportunities to demonstrate their understanding in practical ways.

The course also encourages **peer learning and reflection**, especially through the "code review" component in tutorials. Here, students present and discuss pieces of code, gaining experience in critical thinking and verbal articulation of programming concepts. These reviews not only reinforce learning but also simulate real-world collaborative software development environments.

Finally, **formative and summative assessments** are strategically placed to support learning. Labs provide regular formative feedback, helping students track their progress. Assignments and the final exam serve as summative assessments, challenging students to synthesize their knowledge and apply it to novel problems under realistic constraints.

Together, these methodologies support a deep, active, and structured learning experience, equipping students with both theoretical knowledge and practical skills essential for understanding computer systems.

Course Schedule and Structure

Week	Topic	Overview	Related CLO
1	Course Introduction, bit manipulation, integer representation	Welcome, C Revision, man	1
2	Instruction set architecture and MIPS assembly programming	Bit manipulation, Integers	3,4
3	MIPS assembly programming	MIPS assembly programming	3,4
4	MIPS assembly programming	MIPS assembly programming	4
5	Data representation: IEEE-754; UTF-8	MIPS assembly programming	4
6	File Systems	IEEE-754; UTF-8	4
7	Flexibility Week: No lectures	No Labs or Tutorials	
8	File Metadata	Manipulating files	2,8
9	Processes	Manipulating file meta data and directories	2,6
10	Parallelism, concurrency, synchronisation, coordination, communication	Processes	8
11	Virtual Memory	Concurrency	7
12	Exam Preparation	Virtual Memory, Exam Preparation	5,7

SECTION 3

Assessment details

This section outlines the assessment tasks for this course in detail. Use this information as a reference to guide your approach to completing each task.

Assessment Schedule

Task	Weighting	Due date	Course learning outcomes assessed
1. Assignment 1 (assembly language)	20%	Week 8	4
2. Assignment 2 (C programming)	20%	Week 12	2,5
3. Labs	15%	Weekly	1-7, 9
4. Final Exam	45%	Exam Period	1-9
TOTAL	100%		

All marks will be determined in accordance with the [UNSW College Assessment Policy and Procedure](#). You are strongly encouraged to attempt/submit all assessment tasks within this course.

Programming assignments are marked primarily on their correctness with respect to the assignment specification. Test cases will be provided for each assignment, and further (unseen) test cases will be used for marking. A small component of the mark will be for code quality.

Lab exercises are similarly marked primarily on their correctness, demonstrators will also give feedback on code quality for some labs.

Final Mark

Your final mark for this course will be computed using the above assessments as follows:

CourseWorkMark	=	LabMark + Ass1Mark + Ass2Mark	out of 55
ExamMark			out of 45
ExamOK	=	ExamMark $\geq$ 18.0/45	true/false
FinalMark	=	CourseWorkMark + ExamMark	out of 100

FinalGrade	=	UF, if ! ExamOK && FinalMark $\geq$ 50 FL, if FinalMark $<$ 50/100 PS, if $50/100 \leq$ FinalMark $<$ 65/100 CR, if $65/100 \leq$ FinalMark $<$ 75/100 DN, if $75/100 \leq$ FinalMark $<$ 85/100 HD, if FinalMark $\geq$ 85/100
------------	---	--

Assessment Guidelines

You are responsible for reading, understanding, and following the instructions about (all information below can be found in the UNSW College Assessment Procedure on this web page <https://www.unswcollege.edu.au/about/policies/policy-documents#policies>):

- Examinations
- Assessment tasks other than examinations
- Submission and return of assessment tasks, including rules about late assessment submissions
- Feedback on assessment
- Educational adjustments
- Special considerations including applications for extension and deferred examinations
- Supplementary assessments
- Results

ASSESSMENT WORDCOUNT

- You are required to adhere to the word count specified for each assessment.
- A 10% leeway above or below the word limit is permitted without penalty.
- Exceeding this 10% threshold will result in a penalty deduction.
- Details of word count penalties are provided in the assessment section of this Course Outline.

LATE SUBMISSION/COMPLETION OF ASSESSMENTS

Submitting assessments on time is important to ensure fairness for all students. Penalties will apply for late submissions as follows:

- A 5% penalty will be applied for each day the assessment is late, up to a maximum of five days.
- After five days, you will not be able to submit your assessment unless you have an approved educational adjustment or special consideration.
- Penalties are calculated based on the total marks available for the assessment.
- If you submit an assessment more than five days late without approved educational adjustment or special consideration, you will receive a zero mark for that assessment.

Assignment 1 (assembly language)

Assessment Type	Practical
Weighting	20%
Course learning outcomes addressed	CLO4: Application of established engineering practice (EA2.1)
Type of collaboration	Individual
Task length	No requirement
Due date and time	Week 8 Monday 9.00am
Total Marks	100 marks
AI Category	[ AI tools can be used in an assistive role
Submission Details	Submission through CSE terminal

1. Task Overview

For this assessment, you will complete a practical task in C programming that demonstrates your ability to apply the knowledge gained throughout the course. You will develop, test and debug a C program that incorporates concepts such as memory management, pointers, arrays, data manipulation, control structures and algorithm implementation.

You will also be required to participate in a short face-to-face viva. During the viva, you will briefly present your work and answer questions about your implementation. You may be asked to explain your design decisions, describe how specific sections of your code work, identify how memory is managed, and suggest how your solution could be improved.

The viva will take approximately five minutes and is intended to confirm your understanding of the submitted work. You should be able to explain and defend the code you submit.

THE VIVA WILL CONTRIBUTE 20% OF YOUR TOTAL ASSIGNMENT MARK

2. Rationale

This task is designed to assess your ability to apply C programming concepts in a practical context. It provides an opportunity to demonstrate your understanding of memory management, pointers, arrays, control structures, data manipulation and algorithm design.

The face-to-face viva will also assess your understanding of your own implementation. Explaining and defending your work will help demonstrate that you can reason about your code, justify your decisions and identify potential errors or improvements. These are essential skills for further study and employment in systems programming and software development.

3. Supporting Information

AI CATEGORY: 2 – Generative AI tools may be used to assist your learning and help you understand code, structure or syntax. However, the final submitted work must be entirely your own. If AI tools are used, you must declare their use in your submission.

You must be able to explain and discuss all submitted code during the viva. Use of AI does not remove your responsibility to understand, test or defend your work.

4. Submission Requirements

Submit the following through the CSE servers:

- Your completed .s file
- Any other files specified in the assessment instructions

Your program must be thoroughly tested, appropriately commented and compile and run correctly according to the stated requirements.

You will also attend a scheduled five-minute face-to-face viva, where you will briefly present your solution and answer questions about your code.

Assignment 2 (C programming)

Assessment Type	Practical
Weighting	20%
Course learning outcomes addressed	CL02: Understanding of specialist bodies of engineering knowledge (EA1.3) CL05: Conceptual understanding of computer underpinnings (EA1.2)
Type of collaboration	Individual
Task length	No requirement
Due date and time	Week 12 Monday 9.00am
Total Marks	100 marks
AI Category	[ AI tools can be used in an assistive role
Submission Details	Submission through CSE terminal

5. Task Overview

For this assessment, you will complete a practical task in C programming that demonstrates your ability to apply the knowledge gained throughout the course. You will develop, test and debug a C program that incorporates concepts such as memory management, pointers, arrays, data manipulation, control structures and algorithm implementation.

You will also be required to participate in a short face-to-face viva. During the viva, you will briefly present your work and answer questions about your implementation. You may be asked to explain your design decisions, describe how specific sections of your code work, identify how memory is managed, and suggest how your solution could be improved.

The viva will take approximately five minutes and is intended to confirm your understanding of the submitted work. You should be able to explain and defend the code you submit.

THE VIVA WILL CONTRIBUTE 20% OF YOUR TOTAL ASSIGNMENT MARK

UNSW College

Building L5, UNSW Sydney Campus, 223 Anzac Parade, Kensington NSW 2033 Australia

T: +61 (2) 8936 2222 | W: unswcollege.edu.au

UNSW Global Pty Limited ABN 62 086 418 582 trading as UNSW College™. UNSW College CRICOS Provider Code 01020K. UNSW College TEQSA Provider ID: PRV13020 (Institute of Higher Education)

6. Rationale

This task is designed to assess your ability to apply C programming concepts in a practical context. It provides an opportunity to demonstrate your understanding of memory management, pointers, arrays, control structures, data manipulation and algorithm design.

The face-to-face viva will also assess your understanding of your own implementation. Explaining and defending your work will help demonstrate that you can reason about your code, justify your decisions and identify potential errors or improvements. These are essential skills for further study and employment in systems programming and software development.

7. Supporting Information

AI CATEGORY: 2 – Generative AI tools may be used to assist your learning and help you understand code, structure or syntax. However, the final submitted work must be entirely your own. If AI tools are used, you must declare their use in your submission.

You must be able to explain and discuss all submitted code during the viva. Use of AI does not remove your responsibility to understand, test or defend your work.

8. Submission Requirements

Submit the following through the CSE servers:

- Your completed .c file
- Any other files specified in the assessment instructions

Your program must be thoroughly tested, appropriately commented and compile and run correctly according to the stated requirements.

You will also attend a scheduled five-minute face-to-face viva, where you will briefly present your solution and answer questions about your code.

Labs

Assessment Type	Collaborative
Weighting	15%
Course learning outcomes addressed	CLO 1 Understanding of underpinnings (EA1.1) CLO 2 Understanding of specialist bodies of engineering knowledge (EA1.3) CLO 3 Understanding of specialist bodies of engineering knowledge (EA1.3) CLO 4 Application of established engineering practice (EA2.1) CLO 5 Conceptual understanding of computer underpinnings (EA1.2) CLO 6 Understanding of specialist bodies of engineering knowledge (EA1.3) CLO 7 Understanding of specialist bodies of engineering knowledge (EA1.3)
Type of collaboration	Individual
Task length	No requirement
Due date and time	Weekly
Total Marks	2
AI Category	[ AI tools can be used in an assistive role

Submission Details	Submission through CSE terminal

Task Overview

- 1 Lab Class – 2 hours per week
- **YOU MUST ATTEND AT LEAST 5 GROUP LABS TO PASS THIS UNIT (HURDLE)**
- Attendance will be taken for all Lab sessions.
- Lab classes involve small exercises where students build systems that illustrate the ideas covered in lectures.
- Each Lab session will include short group presentations (2–5 minutes per group) near the end of class. Each group will select one lab exercise to present briefly, highlight any debugging challenges, and explain their learning. These informal, verbal presentations aim to reinforce communication and collaborative problem-solving.
- Each weekly lab (worth 2.0 marks total) will be assessed as follows:
 - 1.0 mark for individual submission (via “give” command)
 - 1.0 mark for active group participation and mini-presentation in Lab. Lab demonstrators will guide group formation and engagement. Students can finish incomplete exercises after class and submit before the deadline. Late penalties apply as per the lab submission policy.
- Optional Challenge Exercises are available for students wishing to extend their learning.
- There will be a Practice Exam towards the end of term in lecture time where you can gain another mark towards your lab mark.
- The total marks for the Practice Exam and Problem sets are capped at 15.
- Students may submit their labs up to one hour late without penalty. For any submission between one and twenty hours late, the grade will be reduced by five marks for every hour (rounded up to the next full hour). Submissions received more than twenty hours late will receive no marks. That means, at the 1 hour mark (exactly), -5 marks, at the 2 hour mark, -10 marks, at the 20 hour mark, -100 (guaranteed zero). Also, the penalty is currently off the ceiling. *No extensions or special considerations will be granted beyond these deadlines.*

The following is a general outline of how the lab activities will be scheduled.

Week	Activity
Week 1	Group Activity
Week 2	Group Activity
Week 3	Group Activity
Week 4	Group Activity
Week 5	Group Activity
Week 6	Group Activity
Week 7	Flex Week, No LABS
Week 8	Assignment Viva
Week 9	Group Activity
Week 10	Group Activity
Week 11	Assignment Viva
Week 12	Consultation/ viva

Rationale

This task is designed to assess your understanding and application of C and MIPS programming concepts. It is a means for students to engage with the material that is presented in lectures in a practical environment. The concepts covered in labs will test students' understanding of the concepts that are taught in lectures. The labs were designed to challenge student thinking and help develop problem-solving skills which can be leveraged throughout their careers as students. Debugging is another key concept that students must master in order to finish all of their lab work.

Supporting information

AI CATEGORY: 2 – generative AI tools may be used to assist your learning and help you understand code, structure or syntax. However, the final submitted work must be entirely your own. If AI tools are used, you must declare their use in your submission.

SUBMISSION: submit your files through the CSE servers every week before the due date. Make sure your code is thoroughly tested, commented, and runs correctly. Include any required README or declarations with your submission.

You are permitted to use the tools `dcc-help` and `dcc-sidekick` to help you understand the error messages you may get when compiling the code you have written.

You are permitted to use `autotest-help` to help you understand why your code may not be passing the automated tests.

You are permitted to use Study Buddy GPT to help you understand why your code may not be passing the automated tests.

You are not permitted to submit code generated by any other AI tools such as Github Copilot, ChatGPT, Google Bard in DPST1091/CPTG1391 for assignments. Submitting code generated by Github Copilot, ChatGPT, Google Bard and similar tools will be treated as plagiarism.

Final Exam

Assessment Type	
Weighting	45%
Course learning outcomes addressed	CLO 1 Understanding of underpinnings (EA1.1) CLO 2 Understanding of specialist bodies of engineering knowledge (EA1.3) CLO 3 Understanding of specialist bodies of engineering knowledge (EA1.3) CLO 4 Application of established engineering practice (EA2.1) CLO 5 Conceptual understanding of computer underpinnings (EA1.2) CLO 6 Understanding of specialist bodies of engineering knowledge (EA1.3) CLO 7 Understanding of specialist bodies of engineering knowledge (EA1.3) CLO 8 Application of established engineering practice (EA2.1)
Type of collaboration	Individual
Task length	No requirement
Due date and time	At the end of term date TBA
Total Marks	100
AI Category	[



AI tools cannot be used

Submission Details

Submission through CSE terminal

Task Overview

There will be a final exam during the exam period. You will be required to attend the exam Face to Face if you are in the country. The exam is closed book you will not have access to your notes. It may contain a mixture of: implementation tasks (which will require you to write C and assembler programs); "theory" questions (which require analysis and written answers); multiple choice questions. During this exam you will be able to execute, debug and test your answers. The implementation tasks will be similar to those encountered in lab exercises. There is a hurdle requirement on the final exam. If you do not score at least 40% on the exam, you cannot pass this course. If your overall course score exceeds 50%, despite scoring very poorly on the exam (<40%), the hurdle will be enforced via a grade of UF.

Rationale

The final exam is a key component of this course, as it provides a comprehensive summary of the concepts, skills, and problem-solving approaches you have developed throughout the term. Successfully passing the exam demonstrates that you have achieved the course learning outcomes and are able to apply your knowledge independently. The exam is conducted under closed-book conditions to ensure a fair and consistent assessment environment for all students, and to provide a balanced counterpoint to other assessments where AI tools may be used. This helps validate your individual understanding and reinforces academic integrity.

Supporting information

The final exam is going to be closed book you will not have access to any notes or the internet.

You will have access to standard linux tools, Mipsy web and DCC-help. More information on this will be given in lectures.

AI CATEGORY: 1 – No Gen AI for this assessment.

SUBMISSION: Submit your work to CSE servers using give. More information will be provided in the exam

SECTION 4

Essential Resources for Students

Essential resources are the materials and technologies you will require to complete this course. The essential resources for this course include:

- Any laptop that can run TigerVNC or VScode

Support for Students

UNSW College offers a range of support services to help you with your studies and student experience:

Resources	
UNSW College Current Students Website	A wide range of resources are available to support your academic and personal success. These include study advice, time management tools, course planning assistance, and academic support services. The Current Student Website also provides quick links to essential information, such as: • Health and Wellbeing resources • Academic support and learning tools • Student news and events • Forms and key administrative services Visit the Current Student Website at https://my.unswcollege.edu.au/ to access everything you need in one place.
Teacher Consultations	Schedule time with your teachers for study advice and subject-related support.
Study Club	Book a place through your online learning platform under Student Consultations to join group study sessions.
Student Advisers	Located at the Student Services Office (Level 1, Building L5), Student Advisers can assist with time management, study skills, course planning, and personal issues affecting your studies. Book appointments online via the Support section of the UNSW College Student Hub.
General Assistance	For help with matters such as accommodation, student activities, or personal concerns, visit the Student Services Office.
Subject Consultations	Times and locations will be provided by your teachers or can be found on the online learning platform.

Take advantage of these services to support your learning and overall success at UNSW College.

References

RECOMMENDED TEXTS

[1] Introduction to Computer Systems: From Bits and Gates to C and Beyond, by Yale N. Patt and Sanjay J. Patel; McGraw Hill

[2] The Elements of Computing Systems: Building a Modern Computer from First Principles, by Noam Nisan and Shimon Schocken; MIT Press (web site, including lecture slides)

