ECE2071

Computer organisation and programming

This unit provides an introduction to computers and CPU organisation, assemblers and compilers, and algorithm design for engineering problems. It covers the language C and its implementation on a typical computer, including standard data types, arrays, control statements, functions, including ways of parameter passing, C library functions, pointers, strings, arrays of pointers, structures, linked lists and binary tree data structures, dynamic memory allocations, and calls to assembly language programs. Object-oriented programming is introduced. Software engineering is covered as the methodology of software development and lifecycle models. Operating system concepts are introduced.

Mode of Delivery	Onshore
Workload	3 hours lectures, 3 hours laboratory and tutorial classes and 6 hours of private study per week
Prohibitions	CSE1301, TEC2041, TEC2042, TEC2171, TRC2400
Prerequisites	ENG1060
Chief Examiner	Professor Arthur Lowery
Unit Coordinator:	Professor Tom Drummond
Campus:	Clayton
Phone:	9905 5682
Email:	tom.drummond@monash.edu
Office hours:	Send an email for an appointment
Other Academic	
Campus:	
Phone:	
Email:	
Office Hours:	*Send an email for an appointment*
Campus Coordinator	Ms Melanie Ooi Po-Leen
Campus:	Malaysia
Phone:	5514 6238
Email:	melanie.ooi@eng.monash.edu.my
Consultation hours:	*Send an email for an appointment*

SEMESTER 1
2012

More details available on Moodle accessible via
https://my.monash.edu.au/

ACADEMIC OVERVIEW

Learning Objectives

To understand the basic concepts of computer programming, and to learn to program in the C language.

Graduate Attributes

Monash prepares its graduates to be:

1. responsible and effective global citizens who:
   a. engage in an internationalised world
   b. exhibit cross-cultural competence
   c. demonstrate ethical values
2. critical and creative scholars who:
   a. produce innovative solutions to problems
   b. apply research skills to a range of challenges
   c. communicate perceptively and effectively

Engineers Australia stage 1 competencies

The Engineers Australia Policy on Accreditation of Professional Engineering Programs – requires that all programs ensure that their engineering graduates develop to a substantial degree the stage 1 competencies. Listed below are the activities in this unit that will help you to achieve these competencies.

Note: that not all stage 1 competencies are relevant to each unit.

Stage 1 competencies	**Activities used in this unit to develop stage 1 competencies**
PE1.1 Knowledge of science and engineering fundamentals	Theoretical lecture material, prescribed texts and recommended reading, tutorial problems and laboratory preliminary work.
PE1.2 In-depth technical competence in at least one engineering discipline	To be able to take a written specification and to turn it into a tested and functioning program in either assembler code, C or a PLC ladder diagram.
PE1.3 Techniques and resources	The use of reference material such as textbooks as resources for providing additional information is emphasized.
PE1.4 General knowledge	
PE2.1 Ability to undertake problem identification, formulation, and solution	The writing of computer programmes involves the steps of understanding, or identifying the problem, formulating an approach and then implementing the

	intended approach. Of course a vital final step is to thoroughly test the solution.
PE2.2 Understanding of social, cultural, global, and environmental responsibilities and the need to employ principles of sustainable development	
PE2.3 Ability to utilise a systems approach to complex problems and to design and operational performance	All forms of computer programming are presented in terms of a systems approach involving software engineering concepts of modularity, hierarchy and encapsulation.
PE2.4 Proficiency in engineering design	Many of the tutorial problems and laboratory exercises concern engineering design. In this context this requires taking a specification and turning it into a functioning computer program.
PE2.5 Ability to conduct an engineering project	Many laboratory exercises are formulated as design projects and require an associated principled engineering approach to their solution.
PE2.6 Understanding of the business environment	
PE3.1 Ability to communicate effectively, with the engineering team and with the community at large	
PE3.2 Ability to manage information and documentation	The importance of program documentation is stressed.
PE3.3 Capacity for creativity and innovation	
PE3.4 Understanding of professional and ethical responsibilities, and commitment to them	Strong emphasis is placed on concepts of intellectual property and the ethics of using the work of others
PE3.5 Ability to function effectively as an individual and in multidisciplinary and multicultural teams, as a team leader or manager as well as an effective team member	Laboratory and tutorial exercises are mainly conducted on an individual basis to ensure that each person fully understands the basics of computer programming.

PE3.6 Capacity for lifelong learning and professional development PE3.7 Professional attitudes	To some extent the material presented is open-ended and students are encouraged to perform self-directed learning which is a necessary skill for effective lifelong learning

Assessment Summary

Assessment Task	Value	Due Date
1. Laboratory exercises (marking via Moodle quizzes)	10%	Weeks 2 to 11
2. Class test	10%	Week 7
3. Lab test	10%	Week 12
4. Written examination	70%	Examination period

Teaching and Learning Method

- Practice in the design and analysis portions of the lecture material are given in the tutorial problems
- Laboratory exercises provide the opportunity for analysing problem definitions and then turning them into functioning computer programs. The laboratory exercises illustrate applications of computer programming to electrical engineering.

Laboratory allocation:

There are 2-hours of practice/ laboratory classes scheduled each week, commencing in week 2. Students must enrol in one laboratory class only using Allocate Plus. Students not allocated to a particular laboratory class will not be accepted into that session. Once a particular session is full, no more students can be accepted because of the limited amount of laboratory equipment.

Tutorial allocation:

There is 1-hour of tutorial classes scheduled each week, commencing in week 2. Students must enrol in one session only using Allocate Plus. Students not allocated to a particular class will not be accepted into that session. Once a particular session is full, no more students will be accepted.

Communication, participation and feedback

- Monash aims to provide a learning environment in which students receive a range of ongoing feedback throughout their studies. In this unit it will take the form of group feedback via practice classes, individual feedback, peer feedback, self-comparison, verbal and written feedback, discussions in class, as well as more formal feedback related to assessment marks and grades. Students/You are encouraged to draw on a variety of feedback to enhance their/your learning

Feedback

Our Feedback to you

- Marks from online quizzes
- Self-assessment of answers to tutorial questions
- Marking of laboratory exercises
- Marking of class test
- Marking of lab. test
- Mark given for final written examination

Your Feedback to Us

Monash is committed to excellence in education and regularly seeks feedback from students, employers and staff. One of the key formal ways students have to provide feedback is through SETU, Student Evaluation of Teacher and Unit. The University's student evaluation policy requires that every unit is evaluated each year. Students are strongly encouraged to complete the surveys. The feedback is anonymous and provides the Faculty with evidence of aspects that students are satisfied and areas for improvement.

For more information on Monash's educational strategy, and on student evaluations, see:
http://www.monash.edu.au/about/monash-directions/directions.html

http://www.policy.monash.edu/policy-bank/academic/education/quality/student-evaluation-policy.html

Previous Student Evaluations of this unit

If you wish to view how previous students rated this unit, please go to
https://emuapps.monash.edu.au/unitevaluations/index.jsp

Examination material or equipment

The end of semester written examination for this unit will be 'closed book'. The use of calculators is NOT allowed and there are no other specifically permitted items.

Unit schedule 2011

Week	Lecture	Prac (Tutorial)	Lab	Assignment activity
1 **27Feb**	1. Overview of C program: variables, data types, arithmetic in C, assignment arithmetic combination, unary operators, precedence in arithmetic. 2. Selection construct: conditions, relational operators, logical operators, if_.else statement, nested if statement, multiple-alternatives decision, switch statement.	No tutorial in week 1	No lab in week 1	
2 **5March**	3. Repetition-Iteration: pre-test loop: while_, post-test loop: do .. while, for loops, nested for loops, break and continue statements 4. Modular programming and functions: test strategies,	Tutorial 1	Laboratory Exercise 1 (C programming)	

	defining functions, function prototypes, passing data: pass by value, pass by reference, returning a value, functions for character input			
3 **12March** **Monday Labour Day**	5. Arrays: declaration of arrays, array initialisation, passing arrays and array elements to functions, addresses of arrays. Pointers: use of pointers, declaring pointers, assigning addresses to pointers, indirect addressing concepts	Tutorial 2	Laboratory Exercise 2 (C programming)	
4 **19March**	6. String: Initialising strings, null character, string functions. Arrays and pointers, pointer arithmetic, pointers to arrays in functions, pointers and strings. 7. Recursion: concept of calling a function within the same function. Structure: user defined structure type, structures and functions.	Tutorial 3	Laboratory Exercise 3 (C programming)	
5 **26March**	8. File I/O: sequential file handling in C, basic file I/O functions 9. Dynamic data structures: dynamic memory allocation . Linked lists, stacks, binary tree structure.	Tutorial 4	Laboratory Exercise 4 (C programming)	
6 **2April** **Friday Holiday**	10. Other languages: programmable logic controllers and ladder diagrams	Tutorial 5	Laboratory Exercise 5 (C programming)	
Easter break 6-13 April				
7 **16April**	11. PLC logical & decision making instructions, supporting procedures, **Class Test**	Tutorial 6	Laboratory Exercise 6 (PLC programming)	
8 **23Apr**	12. An Introduction to Computer Abstractions & Technology; Instructions: Language of the Computer	Tutorial 7	Laboratory Exercise 7 (PLC programming)	
9 **30Apr**	13. Communicating w/people, MIPs addressing mode, Computer arithmetic 1/2	Tutorial 8	Laboratory Exercise 8 (Assembly language)	
10 **7May**	14. Computer arithmetic 2/2, The Processor: Datapath & Control 1/2	Tutorial 9	Laboratory Exercise 9 (Assembly language programming	

			and emulation)	
11 **14May**	15. The Processor: Datapath & Control 2/2, Compilers, Linkers & Assemblers 1/2	Tutorial 10	Catch up laboratory for Friday groups	
12 **21May**	**16.** Compilers, Linkers & Assemblers 2/2	Revision	**Lab Test**	
	SWOT VAC			
	Examination period			LINK to Assessment Policy: http://www.policy.monash.edu/policy-bank/academic/education/assessment/assessment-in-coursework-policy.html

ASSESSMENT REQUIREMENTS

Assessment Tasks

Participation
Laboratory attendance for this unit is compulsory

Assessment Task 1: Moodle laboratory quizzes
Due Date: Preliminary quizzes must be completed before midnight of the day before the associated laboratory period. To be allowed to take the completion quiz a student must have completed the preliminary quiz and demonstrate completion of the associated laboratory exercise.

Guarantee

I can guarantee that occasionally circumstances will arise that will prevent you from completing quizzes at the time you intended. For this reason I recommend that you complete them early. By early I mean at least one day early so that if your computer breaks down, your Internet connection dies, etc., etc. you will still complete the quiz by the due time.

Because I have given you this guarantee that I fully expect some “unexpected” circumstance to interfere with you completing the quiz this will not be accepted as a reason for giving special consideration for missing a quiz deadline.

However, special consideration will be given in cases involving serious medical and other similar circumstances providing adequate documentation is provided as soon as possible after the event.

Details of task:
The quizzes require general knowledge about laboratory exercises as well as knowledge specific to the associated laboratory exercise.

Value:
All quizzes are of equal value.

Assessment Task 2: Class test
Date: A written test to be taken during normal lecture time in week 7
Details of task: A test of your knowledge of C programming (closed book, no calculators).
Value: 10% of your overall assessment.

Assessment Task 3: Lab test
Date: During your normal laboratory period in week 12.
Details of task: You are to demonstrate your ability to write a program in MIPS assembler. This test will be marked on-the-spot.
Value: 10% of your overall assessment.

Examination

The written examination for this unit is worth 70% of your overall assessment.

Returning assignments

The Class Test will be available for collection and marks posted on Moodle within two week of sitting the test.

OTHER INFORMATION

Policies

Monash has educational policies, procedures and guidelines, which are designed to ensure that staff and students are aware of the University's academic standards, and to provide advice on how they might uphold them. You can find Monash's Education Policies at: http://policy.monash.edu.au/policy-bank/academic/education/index.html
Key educational policies include:

- Plagiarism (http://www.policy.monash.edu/policy-bank/academic/education/conduct/plagiarism-policy.html)
- Assessment (http://www.policy.monash.edu/policy-bank/academic/education/assessment/assessment-in-coursework-policy.html)
- Special Consideration (http://www.policy.monash.edu/policy-bank/academic/education/assessment/special-consideration-policy.html)
- Grading Scale (http://www.policy.monash.edu/policy-bank/academic/education/assessment/grading-scale-policy.html)
- Discipline: Student Policy (http://www.policy.monash.edu/policy-bank/academic/education/conduct/student-discipline-policy.html)
- Academic Calendar and Semesters (insert URL);
- Orientation and Transition (insert URL); and
- Academic and Administrative Complaints and Grievances Policy (http://www.policy.monash.edu/policy-bank/academic/education/management/complaints-grievance-policy.html)

Student Services

The University provides many different kinds of support services for you. Contact your tutor if you need advice and see the range of services available at www.monash.edu.au/students

The Monash University Library provides a range of services and resources that enable you to save time and be more effective in your learning and research. Go to http://www.lib.monash.edu.au or the library tab in my.monash portal for more information.

Students who have a disability or medical condition are welcome to contact the Disability Liaison Unit to discuss academic support services. Disability Liaison Officers (DLOs) visit all Victorian campuses on a regular basis

- Website: http://adm.monash.edu/sss/equity-diversity/disability-liaison/index.html;
- Telephone: 03 9905 5704 to book an appointment with a DLO;
- Email: dlu@monash.edu
- Drop In: Equity and Diversity Centre, Level 1 Gallery Building (Building 55), Monash University, Clayton Campus.

Prescribed text(s) and readings

Deitel H., Deitel P., "C: How to Program 6th Edition", Prentice-Hall, 2010.

Recommended text(s) and readings

D.A. Patterson and J.L.Hennessy, _Computer Organisation: the Hardware/Software Interface_, Elsevier Morgan Kaufmann, 2005 edition or later.

J.R.Hanly and E.B. Koffman, _C program design for Engineers_, 2nd edition , Addison Wesley, 2001.
K.Kruse, C.L.Tondo and B. Leung, _Data structure and program design in C_, 2nd edition 1997.
Johnsonbaugh and Kalin, M.,_Application Programming in ANSI C_, MacMillan, 1993
Bronson, G.J _ C++ for Engineers and Scientist , PWS publishing, 1999.
J.E. Ridley, _Introduction to programmable logic controllers : the Mitsubishi FX_, Arnold , London, 1997.
W. Bolton, _Mechatronics: Electronic Control Systems in Mechanical and Electrical Engineering_, 2nd edition, Addison Wesley Longman Ltd, 1999