Course Outline

COMP1921
Data Structures and Algorithms
Teaching Period 2 2009

Contents

- Course Aims
- Student Learning Outcomes
- Assumed Knowledge
- Course Staff
- Course Details
- Resources for Students
- Course Schedule
- Teaching Rationale
- Teaching Strategies
- Assessment
- Academic Honesty and Plagiarism
- Course Evaluation and Development

Course Aims

This course aims to develop C programming proficiency in students who are not undertaking a Computing major. The course builds on the C programming foundation provided in COMP1911 by introducing common data structures and some more advanced programming techniques. Applications of these techniques are introduced.

Student Learning Outcomes

After completing this course, students will:

- be able to write C programs to specification using simple array, list and tree data structures and algorithms
- be confident C programmers
- understand in broad terms the structure and operation of computer systems
- be able to make use of files for storing and retrieving data
- know how to test and debug C programs
- be able to read and understand C code

This course contributes to the development of the following graduate attributes:

Graduate Attribute	Where Acquired
an in-depth engagement with relevant disciplinary knowledge in its interdisciplinary context	Lectures, Assignments
the capacity for analytical and critical thinking and for creative problem solving	Lectures, Tutorials, Labs, Assignments

Assumed Knowledge

Before commencing this course, students should:

- have completed a first course in C programming
- be able to design, construct, document and test simple C programs that accurately comply with a given specification
- have acquired some familiarity with the Unix/Linux operating system and tools, the C compiler, and the C debugger

These are assumed to have been acquired in COMP1911 after 2007. Students who passed COMP1911 prior to 2008 should enrol in COMP1927 - Higher Data Structures and Algorithms for their second programming course.

Course Staff

Contact Details

Staff Name	**Role**	**Email**	**Phone**
Oliver Diessel	Course Convenor	odiessel at cse.unsw.edu.au	57384
Mei Cheng Whale			

Course Admininstrator	meicheng at cse.unsw.edu.au	55683	

Consultations

Staff Name	Weeks	Day/Time	Location
Mei Cheng Whale	1-2	Monday 13:00-14:00	K17 204
Mei Cheng Whale	1-2	Wednesday 13:00-14:00	K17 204
Oliver Diessel	1-13	Wednesday 14:00-16:00	K17 501B

Course Details

Course Code: COMP1921
Course Title: Data Structures and Algorithms
Units of Credit: 6
Course WebSite: http://www.cse.unsw.edu.au/~cs1921/
Handbook Entry: http://www.handbook.unsw.edu.au/undergraduate/courses/2009/COMP1921.html

Resources for Students

The recommended text for this course is:

> *Alistair Moffat, Programming, Problem Solving, and Abstraction with C, Pearson Educational, Australia, 2003, ISBN 1-74103-080-3.*

The following books might also serve as additional reference material:

> *Brian W. Kernighan and Dennis M. Ritchie, The C Programming Language, 2nd edition, Prentice Hall, USA, 1988, ISBN 0-13-110370-9.*
>
> *Paul Davies, The Indispensable Guide to C, Addison Wesley, England, 1995, ISBN 0-201-62438-9.*

Course Schedule

Week	Lectures	Ch.	Deliverable
Â 1	Introduction, review arrays, memory structure	Â 7	Â
Â 2	Numeric representations, review pointers	Â 6	First lab
Â 3	Computer organization, review program development techniques	Â –	Â
Â 4	Structures	Â 8	Practice exam P1Â
Â 5	Dynamic memory allocation, introduction to linked lists	10	Â
Â 6	Linked lists	10	Â
Â 7	Stacks & queues	10	Â
Â 8	Binary search trees	10	Assignment 1 due
Â 9	Simulation	9	Practice exam P2
10	Approximation techniques, floating point representation, numerical errors	9	Â
11	File operations	11	Â
12	Review	Â –	Assignment 2 due
13	—	Â –	Final lab, Practice exam P3

Teaching Rationale

The following rationale has been used in the course design:

- The best way to learn to program is through practice
  - We have 4 lecture hours per week which will include 2 hours of demonstrations
  - We have 3 hours of tutorial and laboratory exercises per week; lab attendance is compulsory
  - We have 2 assignments to help practice and extend the skills you have been learning about
- We are serious about producing programmers
  - 100% of the marks in this course are earnt for your programming efforts
- We will try to help you
  - Through demonstrations during lectures and labs
  - During tutorials
  - In labs
  - At consultations

- By providing you with practice exam questions

Teaching Strategies

COMP1921 includes lectures, tutorials, labs and assignments. Tutorials and labs are combined into 3-hour tute-lab periods. All components are designed with the aim of improving your ability to program.

Lectures

Lectures are scheduled into a 2-hour block on Tuesdays and a 2-hour block on Thursdays. Roughly half of the time per lecture will be spent explaining theory and programming concepts. The rest of the time will be spent on programming solutions to problems.

The material presented during lectures will not necessarily be comprehensive. Students are advised to read the course textbook to augment the ideas being presented - the depth and breadth of information presented in the text and references of a similar standard should be used as a guide for your learning.

Should you have any questions on the content or presentation, please don't hesitate to ask during lectures. Usually other people will have similar questions.

Tutorials

Tutorials commence in Week 2.

Tutorials offer a vital chance for you to discuss problems and their solution in more depth with talented programmers. Please make the most of the opportunity by preparing for them and engaging in the discussion.

We will make tutorial questions available by 5 PM on the Friday of the week before they are to be discussed. In general, tutors will not be providing answers to the tutorial questions directly. Their role is to facilitate the tutorial by steering the direction of the solution development and presentation. We expect students to participate in finding and presenting solutions to tutorial questions.

Labs

Labs will commence in Week 2 as well. Lab exercises will be released by 5 PM on Friday of the week before they are to be attempted.

It is important that you gain sufficient programming proficiency during the labs to be able to pass the final exam.

Selected lab exercises will be marked during the lab session they are scheduled for. **Late labs will not be accepted.** A total of 10% of the marks in the course will be awarded for completing the selected exercises satisfactorily. The mark for an exercise will be determined on the basis of correctness, programming style and quality of comments. Your best 10 exercise marks will be used to obtain your lab mark.

Assignments

This course involves *two* programming assignments worth a total of 24% of the marks in the course. These are to be completed *individually*. Please pay particular attention to the Plagiarism rules. It is likely that some aspects of the assignments will be tested during the final exam.

Assignments will be released at least 3 weeks before they are due, and submissions will be required by 5 PM Friday of the week in which they are due.

Late Submission: The maximum mark achievable for an assignment will be reduced by 10% for each 24 hour period it is late. Assignments will not be accepted after 5 late periods have expired.

Exam

There will be a final exam worth 60% of the marks in this course. The exam will be a 3 hour on-line exam involving answers to practical and theory questions. Your ability to understand and debug programs as well as your ability to create solutions to programming problems will be tested.

Practice exam problems

During the course, you will be given **three** *Practice Exam Problems* during your regular lab session. These are intended to provide you with experience of the type of programming problems you will need to solve during the final practical exam. These will be scored and contribute 6% to your final mark in the course.

Assessment

Your mark in this course will be determined by the score you obtained in your assignments, labs, practice exam problems and exam, plus any bonus marks. Note that assignment marks will be reduced if they are submitted late and labs will not be accepted late.

The mark in this course is computed as follows:

```
labs   = mark for 10 best labs     (out of 10)
ass1   = mark for assignment 1     (out of 12)
ass2   = mark for assignment 2     (out of 12)
prac   = total for practice exams  (out of  6)
exam   = mark for final exam       (out of 60)
bonus  = total bonus marks         (up to   4)

okExam = (exam >= 24/60)           (after scaling)

mark   = labs + ass1 + ass2 + prac + exam + bonus  (capped at 100)

grade  = HD|DN|CR|PS  if mark >= 50 && okExam
       = FL           if mark < 50 && okExam
       = UF           if !okExam
```

Academic Honesty and Plagiarism

Plagiarism is defined as "using the words or ideas of others and presenting them as your own". UNSW and CSE treat plagiarism as academic misconduct, which means that it carries penalties as severe as being excluded from further study at UNSW. There are several on-line sources to help you understand what plagiarism is and how it is dealt with at UNSW:

- Learning Centre: Plagiarism and Academic Integrity
- MyUNSW: Plagiarism and Academic Misconduct
- CSE: Addendum to UNSW Plagiarism Guidelines
- CSE: Yellow Form (whose terms you have agreed to)

Make sure that you read and understand these. Ignorance is not accepted as an excuse for plagiarism.

The following extract from Richard Buckland's 06s1 offering of COMP1911 summarizes the approach we intend taking in this course very nicely:

> *"All work submitted for assessment must be entirely your own work. We regard unacknowledged copying of material, in whole or part, as an extremely serious offence.*
>
> *In this course submission of any work derived from another person, or soley or jointly written by and or with someone else, without clear and explicit acknowledgement, will at the very least result in automatic failure for the course and a mark of zero for the course. Note this includes including unreferenced work from books, the internet, etc.*
>
> *Do not provide or show your assessable work to any other person. Allowing another student to copy from you will, at the very least, result in zero for that assessment. If you knowingly provide or show your assessment work to another person for any reason, and work derived from it is subsequently submitted you will be penalized, even if the work was submitted without your knowledge or consent. This will apply even if your work is submitted by a third party unknown to you. You should keep your work private until submissions have closed.*
>
> *If you are unsure about whether certain activities would constitute plagiarism ask us before engaging in them(!)*
>
> *Copying without consent, severe, or second offences will result in automatic failure, exclusion from the university, and possibly other academic discipline.*
>
> *These are not idle threats, we search the internet and use plagiarism detection software and a range of search engines to hunt for non-original work...*
>
> *Note that we have experienced cases of plagiarism where the code has been copied from printouts or floppy disks/CDs/USB sticks that have been lost in the lab or stolen from the computer or printer.*
>
> *Generally it is your responsibility to prevent other students from accessing your files, but if you lose work in this way, email your tutor immediately."*

Course Evaluation and Development

This course was developed in response to the difficulties meeting the differing needs, aspirations and capabilities of Computing and non-Computing students within our first programming courses. We have targetted this course at students who require a more practical and less theoretical background to programming for their careers.

At the end of the course, we would appreciate your feedback on your experiences via the University's CATEI system. The School uses the feedback you provide to improve the course for future sessions. The University uses the results of your evaluation to gauge its performance.

Previous suggestions we have attempted to address in 09s2 include: clarifying the course objectives following discussions within the Schools of CSE, EET & SISTM, more extensive use of examples, less emphasis on computer organization, the need to improve course pass rates, the need to better integrate COMP1911 and COMP1921.

If you have any suggestions or concerns about the course during session, please do not hesitate to contact the lecturer, your tutor or the course administrator to let us know.

We hope that you enjoy the course!