Course Information for Session 2, 2003

The present document is available online at

http://www.cse.unsw.edu.au/~cs1011/admin/outline.html

It will be updated from time to time. Check the web regularly for the latest version.

Course Overview

This course will introduce you to computer programming and basic elements of computer science. This includes the design, implementation, and testing of programs as well as reasoning about programs, elementary algorithms, and basic data structures. Moreover, you will learn how to use the Unix programming environment. Programming is explained at the example of the functional language Haskell, which exhibits basic concepts of programming languages in a particularly clean way and allows you to solve relatively complex programming problems after a couple of weeks. The course includes many practical exercises.

Course Objectives

After successfully completing this course, you are able to

- design and implement small programs,
- distinguish between well-written and poorly written programs,
- reason about simple programs,
- understand the basic concepts of algorithms and data structures, and
- use the Unix environment.

Keeping Informed

Important notices related to this course will be displayed on the *course home page*, which is located at

http://www.cse.unsw.edu.au/~cs1011/

It is your responsibility to check this page regularly. Sometimes urgent information may also be sent to you by email. Make sure to check your email daily and pay careful attention to any course-related email you receive. For the email policy, see under the heading Email below.

Staff

Name	Role	Notes
Aleksandar Ignjatovic	Lecturer-In-Charge	Will take the majority of lectures
Jane Brennan	Lecturer and Course Administrator	Will take the remaining lectures and Responsible for the administration of the course
Kheng Guan Kua	Course Programmer	Responsible for Assignment Programming
Zhicong Leo Liang	Course Programmer	Responsible for Assignment Programming

Tutorial and Laboratory Class Allocation

All students attend one 2.5 hour combined tutorial/lab each week. This is a one hour tutorial followed by a one and a half hour lab.

You should have registered for a tutorial via NSS when enrolling for the course. If you experience any problems with your enrollment including your tutorial allocation, please attend an admin consult.

Consultations

If you need to see someone other than your tutor for help, you should attend a consultation. Consultations are run most weekdays in the consultation rooms next to the CSE school office in K-17. The timetable is posted on the window of the consultation room, and also on the course homepage.

There are two types of consultation: *admin* and *teaching*. Select the right type for your problem.

Admin consultations:
For all administrative problems (e.g., changing tutorial times, problems with marks, special considerations) you may attend an admin consult and see the course administrator. Or you can email the course administrator if you don't need to see them face to face.

Teaching consultations:

Starting Week 4 there are daily consultations to help students with any problems in understanding the content of the course. If you have problems or questions which arise after your tute, or if there was not enough time in your tute-lab to address all your questions, you may attend any teaching consultation and see the tutor on duty.

Getting Help

From time to time, various problems may arise in your study of this course. Below is a list of typical problems, with suggestions for where you might seek help with them:

Problem	Where to get help
Troubles understanding course	Your tutor (during tute or by email), OR A consultation tutor
Problem installing software on home PC	
Stuck with assignments, lab work	Your tutor (during tute or by email), *(Consultation tutors will not do your assignments for you!)*
Problem with workstations in Lab	Help Desk (Mec Eng undercroft)
Problem with your login account	
Want to change lab-tute class	Course web page
General administrative problems	Course Administrator *Email*: cs1011@cse.unsw.edu.au *In person*: at an admin consult (timetable on course homepage)
Cannot make it to mid session exam.	
Problems concerning the course web pages	
Been very sick => Assignment late	Student Centre & Course Administrator
Been very sick => Affected final exam performance	Student Centre
Problem with Dial-Up-Access from home	UDUS: DIS <> Connect Helpdesk (9385 1777)
Problems with enrolment in course	CSE School Office (K17 ground floor), or Student Centre
Problems with your course/degree	CSE First year coordinator
Problems coping with university generally	University Counselling Service: extension 5418 (or 9385 5418)
Lost property	Campus Security: extension 6000 (or 9385 6000) CSE School office (K17 ground floor)
Emergency	Campus Security: extension 6666 (or 9385 6666)

Email

You may also ask questions by email; although given the amount of mail we normally get, it may take a while before you get a reply. Since the course is so large we have created several email addresses you can write to, one for each main area of the course. Please be careful to send your mail to the correct address (each address goes to a different person). Do **not** email Aleks Ignjatovic, Jane Brennan, Kheng Guan Kua, or Zhicong Leo Liang directly; instead use the addresses below. Any direct email to the aforementioned persons may be entirely ignored.

Questions about	**Who to email**
Material you don't understand	Your tutor, or visit a consultation.
Problems with marks or marking	Your tutor
Assignment 1	*cs1011.ass1@cse.unsw.edu.au*
Assignment 2	*cs1011.ass2@cse.unsw.edu.au*
All other administrative matters	*cs1011@cse.unsw.edu.au*

Important: All course-related email will be sent to your *CSE email address.* Moreover, teaching staff is advised to only answer course-related email queries that originate from CSE email addresses. Email from other sources will be ignored due to privacy concerns. Failure to read your email does not constitute a sufficient reason for any form of special consideration.

Higher Computing 1A

No Higher Computing 1A stream is offered in Session 2.

Lectures

The purpose of lectures is to introduce you to the concepts covered by this course, show where they fit in the overall scheme of things and provide motivating examples to help you understand them. All assessable material that is covered in the lectures is either discussed in the set textbook or in additional lecture notes. Both texts may contain additional information about the covered topics. You need to read and understand the set textbook and any additional lecture notes to fully cover the course content. The reading guide on the course webpage will help you to know what chapters of the book you need to read, as some chapters might be taught in a different order to the textbook.

Learning research shows that students learn more efficiently when they do the reading *before* attending the lecture. You should attend each lecture. The lecture times and locations are as follows:

Monday	2-3, CLB 7
Tuesday	2-3, MAT A
Wednesday	2-3, KBT

You need to attend all three lectures.

Assessment Summary

The assessable components of the course are the following:

Item	Weight
Tute/Lab mark	10%
Assignment 1	20%
Assignment 2	20%
Final exam	50%
Total	100%

Your exam mark must be a pass in order to pass the course; that is, no student with less than 50/100 in the final exam will be allowed a passing grade in this course.

The final mark will be the **harmonic mean** between the exam mark and the marks from the practical assessments (tute-lab mark and the assignment marks). Where assignment marks are significantly higher than your exam marks, you may be required to attend an interview to explain the difference. Marks may be scaled to ensure that the Pass/Fail boundary and the Distinction/High Distinction boundary are at a consistent standard from session to session.

Special Consideration

Students whose exam performance is affected by **serious** and **un-foreseeable** events outside their control can apply at the student centre for *special consideration*. Special considerations will only be given where **all** other components of the course (assignments, tutes, exams, and labs) have been satisfactorily completed.

The document "Important Advice for Students" states the supplementary assessment policy for CSE. The following criteria should be followed closely when considering a student for a supplementary exam:

- the student has submitted a fully documented request for Special Consideration within 3 days of the exam; and
- the medical or other evidence is clearly of a nature to affect the student's performance seriously (minor ailments like a cold, a headache - unless chronic, or severe enough to require prescribed medication - feeling 'unwell', etc do not excuse a poor result); and
- the student's performance during the semester, particularly in any mid-session test has been of satisfactory standard. Students who fail such a test will not be granted further assessment. Assignment and laboratory marks may not carry much weight in determining whether further assessment is warranted; and
- the student did not attend the exam, or attended the exam but was forced to leave through illness.

The document can be found on-line at

`http://www.cse.unsw.edu.au/school/facilities/yellowform.html`

In particular, make up your mind whether your are ill **before** the exam. Once you have sat an exam, no supplementary exam will be granted. Depending on student numbers, supplementary exams may be **oral.** Oral exams will be conducted on the day before and after the date for written supplementary exams.

Tutes-Lab

Practical programming competency is an important objective of this course. The best way to learn programming skills is to practise programming - you do that in that in weekly laboratories and with assignments. Laboratories and tutorials are integrated into this course. The first hour will be conducted in a tutorial room discussing set tutorial questions and preparing for the lab exercises. This will be followed by 1.5 hours in a laboratory.

The tutorial component will give you a chance to clarify ideas mentioned in lectures and to *practise your problem-solving skills* in a small, more personal, class with the assistance of a tutor. Tutorials are designed to help you learn. They are your main forum for asking questions and getting personal assistance. You should make sure that you use them effectively by examining *in advance* the material to be covered, by asking questions, by offering suggestions, and by generally participating.

In the laboratory component of the class you will work through set programming exercises. This will give you a chance to *develop your programming skills* on small, simple examples. The examples have been chosen to highlight particular aspects of programming, and are designed to assist you in your assignments. Your tutor will be there to assist you.

There are marks for tutorial participation and for the completion of the laboratory exercises.

The tutorial questions and laboratory exercises will be placed on the course web page. You will need to read them and prepare in advance.

The assessment of the laboratory exercises will be done in labs by your tutor. Each week's lab exercises will include a core section and many will include an extension section. Unless otherwise stated all questions are to be completed and marked during the lab in which they are given. Lab exercises must be marked face to face with your tutor during the lab, email submissions will not be accepted. The total lab mark is out of 30.

Mid-session Exam

There is **no** mid-session exam.

Assignments

The programming assignments are a very important part of the course. Assignments give you the chance to practise programming on slightly larger problems (compared to the small programs in the laboratory exercises). The aim is for you to practise all stages of the programming process: Start by understanding the specification, design a method to solve the problem, refine this design, implement the design as a complete program, and test that the program meets the specification. Assignments are a very important part of this course; you absolutely *must* make sure that you attempt them yourself. A portion of the final exam may relate to assignment work.

There are two programming assignments; the second of which will be divided into two stages. The provisional schedule is as follows:

- Assignment 1 will be due in Week 6.
- Assignment 2 stage I will be due in Week 10.
- Assignment 2 stage II will be due in Week 12.

The assignments will be released as early as possible to give you the chance to think about them for a longer period of time. However the material required for doing the assignments will only be covered two weeks before the submission deadlines. The assignments are designed in a way that they can be completed completed in two weeks time. But, past students advise that assignments take **far** longer to complete than you at first estimate, so make sure you start them early and allow plenty of time. *You cannot complete a computing assignment in one week.*

Assignment work can be completed on the workstations in CSE labs or on a PC at home. Assignments must be submitted on-line from a school terminal via the `give` command (submissions by email or on floppy disk will be disregarded). It is in your best interests to make regular backup copies of your work and (because of machine loads on deadline days, for example) to complete and submit assignments well before there deadlines. Failure to meet the assignment deadline due to technical problems (computer crash etc) do not constitute a valid reason for special consideration, except where these problems are caused by the School.

If you wish to submit an assignment late, you may do so, but the maximum available mark for late assignments is reduced by 10% per day for up to four days. Assignments that are more than 4 days late will be awarded **zero marks.** So if your assignment is worth 85% and you submit it one day late you still get 85%, but if you submit it two days late you get 80%, and so on.

Assignment extensions are only awarded for *serious* and *unforeseeable* events. Having the flu for a few days, deleting your assignment by mistake, going on holiday, going abroad, work commitments, etc do not qualify. Therefore aim to complete your assignments before the due date in case of last minute illness, and **make regular backups of your work**. No extension will be granted without a fully documented submission for special consideration at the Student Centre.

Plagiarism

All work submitted for assessment must be entirely your own work. We regard copying of assignments, in whole or part, as a *very serious offence.*

In this course, even the attempt to obtain code for a lab exercise or assignment from others (e.g., asking for code on a Haskell mailing list or other public forum) will result, at the very least, in 0 marks independent of whether any code has been submitted for assessment or not.

Actual submission of work derived from another person, or jointly written with someone else will by default result in an **automatic fail** of the course with a mark of **0 FL.** At the disgression of the lecturer a reduced penalty may be applied in the case minor offences. Such a reduced penalty typically results in a **negative** mark in the order of the maximum mark for the assignment (i.e., if an assignment is worth 20 marks, a person submitting work derived from somebody else will receive -20 marks). For repeating offenders, an application for suspension from the University will be filed.

Do not provide or show your assignment work to any other person. Allowing another student to copy from you will, at the very least, result in zero for **your** assignment. If you knowingly provide or show your assignment work to another person for any reason, and work derived from it is submitted, you will be penalised, 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.

Copying without consent, severe, or second offences will result in automatic failure, **suspension from the University**, and possibly other academic discipline.

See also the *Unix Primer* and the Yellow Slip for additional information. If the penalties set out on this page, the Unix Primer, or the Yellow Slip differ for any situation, the more severe penalty applies.

The following list details some of the possible behaviours that constitute plagiarism including the applicable penalty:

Copying of code: 0 FL (for the whole course)
: This includes submitting another students solution to the assignment in full or part and with or without modifications as ones own solution. This includes code that was derived from pseudo code or a detailed algorithmic description.

Help with coding: negative marks (ie, minus the max mark)
: Submission of code that was developed with the help of another student or a third party. This includes receiving pseudo code or discussing details of the solution, or having a third party debug your code.

Originator of plagiarised solution: 0 marks (for this assignment)
: Showing ones own solution to another student or giving a copy to another student, or providing help in the form of pseudo code or detailed algorithmic descriptions. (This does NOT apply if there is sufficient evidence that the copying happened without knowledge of the originator.)

Team work: 0 marks (for this assignment)
: Submission of code that was developed fully or in part with another student. This includes joint development of pseudo code or a detailed algorithmic description. (It does not matter if all parties can explain the code afterwards. The code must not be jointly developed.)

These are no idle threats, we use plagiarism detection software to search for multiply-submitted work. In Session 2, 2001, roughly **50** out of slightly over 600 students failed the course after being caught cheating.

Final Examination

The final examination in this course will be held during the November/December examination period; it will be a closed book exam and examine all material covered in the course. Supplementary examinations will be held in December. It is *your responsibility* to check notice boards, the School web pages, and your email for details of supplementary examinations and to ensure that you are available at the required time. See also the information about special considerations above.

Further Information

There is a large amount of information stored in the course home page, and further will be added over the session. You should explore the course web site and check back regularly for updates.

Text Books

- Manuel M. T. Chakravarty & Gabriele C. Keller, ``*An Introduction to Computing (with Haskell)*'', Pearson Education (2002).
  [This text covers most of the material discussed in the lectures. It is available from the UNSW bookshop.]
- ``*The Concise Unix Primer*'', School of CS&E, UNSW.
  [Take it with you when attending labs; available at the Help Desk.]

Reference Books

- Paul Hudak, ``*The Haskell School of Expression: Learning Functional Programming through Multimedia*'', Cambridge University Press (2000).
  [Very good book, which is recommended if seek additional explanations and background information.]
- Simon Thompson, ``*Haskell, The Craft of Functional Programming*'', Addison-Wesley, Second Edition (1999).
  [Used in earlier instances of this course.]

[an error occurred while processing this directive]