Staff

Staff Name	Role	Email	Extension
Maurice Pagnucco	Lecturer-in-Charge	morri@cse.unsw.edu.au	56925
Rex Bing Hung Kwok	Lecturer	rkwok@cse.unsw.edu.au	

Lecture Times

Day	Time	Room
Tuesday	15:00-16:00	OMB 149
Thursday	12:00-14:00	Webst ThA

Communication

Sometimes urgent information may be sent to you by email. Make sure you pay careful attention to any email you receive.

All official email will be sent to your CSE email address. If you prefer to read your mail at some other address you can redirect your mail using mlalias. Ask your tutor or the help desk, if you need help doing this.

Additional information will be provided in the course MessageBoard. You should read these regularly. These forums are the best place to ask questions about the course.

Pre-requisites

COMP2041/9041 assumes that you have a sound understanding of the C programming language - An understanding equivalent to the coverage in the course prerequisite: one of COMP1021, COMP1721, COMP1911 or COMP9021.

It would also be useful if you had passed COMP1921, COMP2011 or COMP9024 before attempting this course, simply to have a greater level of programming experience.

Aims

This course is designed for students who have mastered the basics of programming. It aims to broaden your knowledge of techniques and tools for software construction.

By the end of the course, you should:

- have practical experience in programming the scripting languages Perl and the Unix shell
- have a broader & deeper knowledge of building software systems in the C programming language
- more appreciation of the use of specific technologies and strategies during software development
- exposure to tools for debugging, performance improvement, configuration and version control

Syllabus

The following is a summary of the topics that will be covered in this course.

1. Tools for software construction
    - Programming languages (C)
    - Scripting languages (Perl, Shell, CGI)
    - Filters (awk, sed, grep, tr, ...)
    - Analysis tools (debuggers, profilers)
    - Configuration tools (make, rcs, cvs)
2. Techniques for software construction
    - Analysis, design, coding, testing, debugging, tuning
    - Interface design, documentation, configuration
3. Qualities of software systems
    - Clarity, reliability, efficiency, portability, ...

The focus for the practical work will be on C and Perl. However, you would be well advised to acquaint yourselves with the facilities provided by the Unix shell.

Lectures

Lectures will be used to present the theory and practice of the techniques and tools in this course. There will be extensive use of case studies and practical demonstrations during lectures. Lecture notes will be available on the course web pages before each lecture.

Tutorials

From Week 3 you will also be expected to attend a one-hour tutorial session to clarify ideas from lectures and work through exercises based on the lecture material. You should make sure that you use them effectively by examining in advance the material to be covered in each week's tute, by asking questions, by offering suggestions and by generally participating. The tutorial questions will be posted on the Web in the week before each tute. There are no marks for tutorial attendance.

Laboratory Classes

Following the tute class each week, there will be a two-hour lab class, during which you will work on a variety of small practical problems involving the tools introduced in lectures. Because this course is practical in nature, lab class are a very important component, and you should make every effort to attend the labs and complete the exercises diligently. In particular, **keep up-to-date** with the Lab work; if you fall behind it affects your ability to understand later material in the course.

To obtain a mark for a lab exercise you must both demonstrate the completed lab exercise to your tutor during a lab class and submit it using give.

If you don't complete a lab exercise during the scheduled class, you can still obtain the mark if you both submit the completed exercises before midnight Sunday and you demonstrate it to you tutor in the follow week's lab.

COMP9041 students are recommended to attend tutorials and labs, but may opt to complete the work in their own time and not have it formally assessed. To do this, they must advise the lecturer by email by the end of Week 2.

Assignments

In the assignments, you will work through the process of building and/or modifying software systems, using the tools and techniques described in lectures. The first assignment will involve Perl, while the second assignment will involve C programming. The assignments together contribute 30% of the overall mark for the course.

Ass	Description	Due	Marks
Stage 1	Perl	Week 8	10%
Stage 2	Perl	Week 10	10%
Stage 3	Perl/CGI	Week 14	10%

The assignments have not been prepared yet, and it may be necessary to vary the weights or deadlines when they have been prepared.

Exam

There will be a three-hour primarily practical exam, to be held in the CSE labs during the exam period. It consists of five small implementation tasks and one written section. 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

You will not be expected to remember the details of programming languages used in the course; reference information will be provided along with the exam paper, giving a summary of any language that we expect you to use.

It is a hurdle requirement for this course that you pass the exam.

It also is a hurdle requirement for this course that you perform satisfactorily on the implementation tasks in the exam. This is defined as successfully at least two of the five implementation tasks.

Assessment

Component	Value
Lab Work	10%
Assignments	30%
Exam	60%

These assessment weights might be varied by a few percent when the assignments have been chosen.

If your final exam mark is less than 50% then your overall mark will not be allowed to exceed your exam mark. In other words you must pass the final exam to pass the course.

As mentioned above, your performance on the practical component of the final exam must also be satisfactory to pass the course.

The lecturer may scale overall marks, or individual components, up or down to obtain a desired mark distribution.

You will not be permitted to sit the prac exam unless you have obtained 12/40 for assignments+labs.

Reading Materials

There is no required textbook for the course. Useful reference books include the following:

- Kernighan & Pike, *The Practice of Programming*, Addison-Wesley, 1998.
  (Inspiration for 2041 - gives theory, philosophy and some tool details)
- Wall et al., *Programming Perl* (3rd ed), O'Reilly, 1996.
  (Invaluable Perl reference manual)
- McConnell, *Code Complete*, Microsoft Press, 1993.
  (Ok, ok, so it's Microsoft Press ... but it has many interesting case studies and good ideas)
- Peek, O'Reilly, Loukides, *Unix Power Tools*, O'Reilly, 1997.
  (Comprehensive guide to common Unix tools)
- Robbins, *Unix in a Nutshell: System Vr4 and Solaris 7*, O'Reilly, 1995.
  (Concise guide to Unix and its toolset)
- Loukides & Oram, *Programming with GNU Software*, O'Reilly, 1997.
  (Concise guide to the GNU toolset (gcc, gdb, make, gprof))
- Sebesta, *A Little Book on Perl*, Prentice Hall, 1999.
  (Neat little book on Perl, if you don't want a big heavy one)
- Christiansen et al., *Perl Cookbook*, O'Reilly, 1998.
  (Lots and lots and lots of interesting example Perl programs)
- Kernighan & Pike, *The Unix Programming Environment*, Prentice Hall, 1984.
  (Pre-cursor to the textbook, intro to Unix tools)

For pointers to other useful reading material, including documentation for all of the tools used in the practical work, see the course Web pages.

Plagiarism

All work submitted for assessment must be your own work. Lab exercises and assignments must be completed *individually*. We regard copying of assignments or lab exercises, in whole or part, as a very serious offence. We use plagiarism detection software to search for multiply-submitted work.

1. Submitting part or all of other students' work, with or without acknowledgement, is not acceptable.
2. Submitting solutions written by other persons is also not acceptable.
3. Building on ideas and partial solutions obtained from public sources, such as web resources, may be acceptable, provided full acknowledgement is made. However, the final mark will take into account the starting point and how much development work would have been required. Failing to acknowledge web or other resources is unacceptable.
4. Discussing approaches to solutions with other students is quite appropriate, but any discussions should remain at the design level, and must not include program text. Comparison tools will detect any common code across the student body.
5. The safest approach is to work diligently on your own, seeking help from the forum or course staff.

- Submission of work derived from another person, or jointly written with someone else will, may result in automatic failure for COMP2041 with a mark of zero.
- Allowing another student to copy from you will may result in a mark of zero for your own assignment or lab exercises. Do not provide your work to any other person, even people who not UNSW students. You will be held responsible for the actions of anyone you provide your work to.
- Severe or second offences will result in automatic failure, exclusion from the university, and possibly other academic discipline.

Refer also to the Yellow Form material on plagiarism and Learning Centre website