Subject Introduction

Software Construction: Techniques and Tools

Aims

This subject aims to explore techniques and software tools that can be used in the production of high-quality software.

By the end of the subject, students should:

- understand the software development process and appreciate the use of specific technologies and strategies during different phases of software development
- be able to implement a software system based on the Unix shell and Perl scripting languages
- be able to implement components of a software system in the C, Java and Haskell programming languages
- be able to choose appropriate interfaces for users to interact with software systems and for software systems to interact with each other
- be able to experimentally analyse properties of developed programs, both in terms of correctness (by testing) and efficiency (by profiling)
- be able to understand the configuration and installation of software systems under the Unix environment that use Make, CVS and Configure scripts

Pre-requisites

It would be preferable if you had passed COMP2011 Data Organisation before attempting this subject, simply to have a greater level of programming experience. However, you should be able to cope ok as long as you have a good pass in COMP1021 Computing 1B. You can do COMP2011 concurrently with COMP2041 if you wish. COMP2021 Digital Systems Structures is not specifically relevant for this subject, but, once again, if you have already done it, the extra experience with programming will help you.

Web Pages

The subject web pages will provide links to all material related to this subject. The pages can be accessed via:

From any CSE lab workstation:	file:/home/cs2041/web/index.html
From anywhere on the Internet:	http://www.cse.unsw.edu.au/~cs2041/

You should become familiar with the structure and content of the subject web pages early on, and consult them frequently. In particular, all important announcements will be posted to the Subject Noticeboard on the opening page.

Staff

Name	Room	Phone	Email	Role
John Shepherd	Elec Eng 325	9385 6494	jas	Lecturer-in-charge
Tim Lambert	Elec Eng 334	9385 6496	lambert	Lecturer
Malcolm Ryan	Samuels Gnd	9385 3988	malcolmr	Tutor/Administration

Syllabus

The following is a summary of the topics that will be covered in this subject. For more information (e.g. level of detail for each topic) consult the subject Web pages.

1. Qualities of software systems
   - Correctness, clarity, reliability, efficiency, portability, ...
2. Techniques for software construction
   - Analysis, designing, testing, debugging, documenting, tuning, configuring
3. Tools for software construction
   - Programming languages (Java, Haskell, C)
   - Scripting languages (Shell, Perl, CGI)
   - Filters (awk, sed, grep, tr, ...)
   - Analysis tools (debuggers, profilers)
   - Configuration tools (make, rcs, cvs)

Reading Material

The following book is the recommended text for this subject:

- Kernighan & Pike, *The Practice of Programming*, Addison-Wesley, 1998.
  (Gives theory, philosophy and some tool details)

Other useful reference books include the following:

- Peek, O'Reilly, Loukides, *Unix Power Tools*, O'Reilly, 1997.
  (Comprehensive guide to common Unix tools)
- Gill & Loukides, *Unix in a Nutshell: System V v2.0*, O'Reilly, 1995.
  (Concise guide to Unix and its toolset)
- Wall et al., *Programming Perl* (2nd ed), O'Reilly, 1996.
  (Invaluable Perl reference manual)
- 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 subject Web pages.

Classes

Each week there will be a two-hour lecture on Thursday during which new material (theory, practical demonstrations and case-studies) will be presented. The Tuesday lecture slot will be reserved for revision and catch-up material, according to student demand.

Lecture timetable:

Day	Time	Location
Tuesday	5pm-6pm	Mathews A
Thursday	3pm-5pm	Keith Burrows Theatre

Each week students 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. Tutorial registration will occur via Sirius in week 2.

Following the tute class each week, there will be a two-hour lab class, during which students will work on a variety of small practical problems involving the tools introduced in lectures. Because this subject is practical in nature, lab class are a very important component, and students should make every effort to attend the labs and complete the exercises diligently.

Assignments

In the assignments, students will work through the process of developing software systems using the tools and techniques described in lectures. The first assignment will deal with text-processing and second with image-processing. The assignments contributes 20% of the overall mark for this subject.

Ass	**Description**	**Due**	**Marks**
1	Text-based application	Week 8	8%
2	Web-based application	Week 13	12%

Assignments are to be completed individually and without assistance from others except the tutorial staff in this subject. Plagiarism will not be tolerated; assignment submissions will be checked for similarity and anyone found guilty of submitting assignment work that they did not do themselves will be awarded a mark of zero for the subject.

Exams

There will be a two-hour practical exam, to be held in the CSE labs on Saturday 30 October. During this exam, students will be required to solve a number of small practical problems, similar in nature to those completed in the lab classes. The prac exam contributes 30% of the overall mark for the subject.

There will also be a two-hour closed-book written exam for this subject in the November exam period. The exam will consist of a mixture of descriptive questions, quantitative analysis, and small database programing exercises, and will include material related to the assignment work. You will not be expected to remember the details of programming languages used in the subject; reference information will be provided on any languages required along with the exam paper. Programs will be marked leniently with respect to to precise syntax. The written exam contributes 40% of the overall mark for the subject.

Assessment

The subject assessment includes components from lab work, project work, a prac exam and a final written exam, in the following proportions:

Component	**Ident**	**Value**		**Component**	**Ident**	**Value**
Lab Work	`lab`	10%		Assignments	`ass`	20%

Prac Exam	`prac`	30%	Final Exam	`exam`	40%

Your final mark in this subject will be based on a combination of the above components. Note that each exam is a hurdle, so that if you fail one of them badly enough, you cannot pass the subject. The following formula describes precisely how the mark will be computed and how the hurdle will be enforced.

```
labs   = mark for lab exercises  (out of 10)
ass    = mark for assignment     (out of 20)
prac   = mark for practical exam (out of 30) [scaled]
exam   = mark for written exam   (out of 40) [scaled]
okExam = prac >= 14/30 && exam >= 16/40

final  = labs + ass + prac + exam, if okExam
       = (prac + exam) * 100 / 70, otherwise
```

John Shepherd, July 1999