Course Outline

COMP2041 / COMP9041
Software Construction

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
- Other Policies and Information

Course Aims

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

Student Learning Outcomes

After completing this course, students will:

- 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

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

Graduate Attribute	Where Acquired
the skills involved in scholarly enquiry	students need to research, compare and evaluate different software construction methods as part of the practical work and assignments
the capacity for analytical and critical thinking and for creative problem solving	laboratory work and assignments requires students to solve a range of problems by choosing and applying appropriate programming tools and techniques
the skills to locate, evaluate and use relevant information (Information Literacy)	students must extensively use online documentation to complete assignment and lab work

Assumed Knowledge

Before commencing this course, students should:

- Have a sound understanding of the C programming language
- Produce a correct procedural program from a specification
- Understand fundamental data structures including char, int, float, arrays, struct,pointers
- Understand fundamental algorithms including sorting and searching
- Appreciate the use of abstraction in computing

For undergraduate (COMP2041) students, these are assumed to have been acquired in in COMP1911 Computing 1 and COMP1921 Algorithms and Data Structures (or COMP1011 Computing 1A and COMP1021 Computing 1B). Undergraduate students may also have already taken COMP2911 Engineering Design 2.

For postgraduate (COMP9041) students, the above material will have been covered in COMP9021 Principles of Programming. Postgraduate students may also have already taken COMP9024 Data Structures/Algorithms, or covered similar material in their undergraduate degree.

Course Staff

Contact Details

Staff Name	Role	Email	Phone
Angela Finlayson	Course Convenor	angf at cse.unsw.edu.au	56852

Consultations

Staff Name	Day/Time	Location
Angela Finlayson	Tuesday 1500-1700	K17-G01
Angela Finlayson	Thursday 1200-1400	Pipe: Stuvac

Course Details

Course Code: COMP2041 / COMP9041
Course Title: Software Construction
Units of Credit: 6
Course WebSite: http://www.cse.unsw.edu.au/~cs2041/
Handbook Entry: http://www.handbook.unsw.edu.au/undergraduate/courses/2009/COMP2041.html

This course is taught in conjunction with

Course Code/Title: COMP9041 Software Construction
Units of Credit: 6
Course WebSite: http://www.cse.unsw.edu.au/~cs2041/
Handbook Entry: http://www.handbook.unsw.edu.au/postgraduate/courses/2009/COMP9041.html

Resources for Students

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 Forums. You should read these regularly. These forums are the best place to ask questions about the course.

Reading Material

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 detail s)
- Wall et al., *Programming Perl* (3rd ed), O'Reilly, 1996.
  (Invaluable Perl reference manual)
- McConnell, *Code Complete*, Microsoft Press, 1993.
  (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.

Course Schedule

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

1. Tools for software construction
   - Scripting languages (Perl, Shell)
   - Filters (sort, sed, grep, tr, ...)
   - Analysis tools (debuggers, profilers)
   - development tools (make, CVS, doxygen, trac)
2. Techniques for software construction
   - Analysis, design, coding, testing, debugging, tuning
   - Interface design, documentation, configuration
3. Qualities of software systems
   - Correctness, clarity, reliability, efficiency, portability, ...

A rough schedule is as follows:

Week	Description
Week 1	Filters
Weeks 2-3	Shell scripting
Weeks 4-6	Perl
Weeks 7-8	Web documents and CGI scripting
Week 9	Testing and Debugging
Week 10	Configuration and building
Week 11	Performance Improvement
Week 12	Version Control

Teaching Rationale

Software construction is a skill that improves with practice, thus this course has a heavy practical orientation. The course is designed in such a way that passing the course will require a good understanding of the fundamental theory as well as good practical skills. Labs and assignments provide great opportunities for developing practical skills, however students will be encouraged to do their own additional self-guided practice, to get maximum results from this course.

Teaching Strategies

- **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. Lectures run from week 1 to week 12.

  Lecture times and locations:

Day	Time	Location
Monday	14:00 - 16:00	Central Lecture Block 2
Tuesday	15:00 - 16:00	Central Lecture Block 8

- **Tutorials:** Tutorials run weekly from week 2 to week 13. You will be expected to attend your allocated one-hour tutorial session each week. Tutorials are used to clarify ideas from lectures and involve working 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 tutorial. There are no marks for tutorial attendance.

- **Lab Work:** Following the tutorial 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:** Assignments are a critical component of the course as they are designed specifically to help you learn, practice and understand the most important material for the course. In the assignments, you will work through the process of building and/or modifying significant software systems, using the tools and techniques described in lectures.

  There will be 3 assignments given, with the focus being on Perl programming. The proposed schedule for assignments is given below. Note: It may be necessary to vary the deadlines slightly.

Ass	Description	Due	Marks
Stage1	Perl	Week 7/8	10%
Stage 2	Perl	Week 10	10%
Stage 3	Perl/CGI	Week 13	10%

Assessment

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

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 all of 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.

Final Mark

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.

You may be excluded from the prac exam if you have < 10/40 for assignments+labs.

Your final mark will be calculated as follows:

```
ass    = mark for assignments     (out of 30)
labs   = mark for assessed labs   (out of 10)
exam   = mark for exam            (out of 60)

okExam = (exam >= 30/60 && two implementation task solved)

mark   = ass + labs + pexam + texam
grade  = HD|DN|CR|PS  if mark >= 50 && okExam
       = FL           if mark < 50 && okExam
       = UF           if !okExam
```

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

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.

Course Evaluation and Development

This course is evaluated each session using the CATEI system. This feedback is used to improve the course materials and their delivery. In recent sessions, no major issues have been reported with the course.

In the previous offering of this courses, some students noted that the workload of some of the labs was too heavy.

After taking into consideration their comments, we have updated some of the lab material.

Other Policies and Information

- Occupational Health and Safety policies
- Information for students with disabilities. Contact the lecturer ASAP if you have any disabilities that may affect this course.