MONASH University

FIT3036
Computer science project

Unit guide

Semester 1, 2009

Last updated : 13 Feb 2009

Table of Contents

FIT3036 Computer science project - Semester 1, 2009

Unit leader :

John Hurst

Lecturer(s) :

Clayton

- Kerri Morgan

Malaysia

- TBA

Introduction

Welcome to FIT3036 Third Year Project (previously offered as CSE3301). It is a 6 credit point unit, offered in first and second semester during 2009. The unit gives you an opportunity to tackle a significantly larger software design and implementation than you have been able to do previously in your course. You will work on a project allocated to you (please remember to identify your preferences) under the supervisor of an academic member of staff. The nature of the project is largely unconstrained, and will be determined in consultation with your project supervisor.

Unit synopsis

This unit is intended to provide practical experience in designing, developing and testing a non-trivial computer science project. Projects are generally software-based, although sometimes they may involve hardware development or investigation of theory. Projects cover the whole process of software (or hardware) development, from analysis through design to implementation and testing. Comprehensive written documentation on the project is required. Students are assigned in groups to a project supervisor. The list of projects to be offered by individual lecturers will be available at the beginning of semester, and students will be assigned to projects in the first week of semester. There are no lectures in this unit, although students will be expected to attend regular meetings with their project supervisor. As a rough guide, this subject should require about one hundred and fifty hours of effort from the student during the semester.

Learning outcomes

Knowledge and Understanding

At the completion of this unit, students will have an understanding of:

K1. Strategies for developing a non-trivial programming, hardware, or theory-based project.

K2. How to locate and utilise prior research and methods on a particular topic.

K3. How to cite bibliographic references the student has used to understand various components of the project, support claims on knowledge, events, hypotheses and theories.

K4. How to document software development from a user and application programming perspective.

K5. Software development methods: analysis, design, implementation and testing applied to the design and development of a non-trivial project.

Attitudes, Values and Beliefs

At the completion of this unit, students will have attitudes that will allow them to:

A1. Acknowledge the importance of attending and contributing to meetings as a method of gaining important information and ideas about the project?

A2. Understand the basic requirements of software development from both user and developer perspectives.

A3. Appreciate the importance of correctly acknowledging the work of others in researching solutions to problems.

A4. Value the role of work books in documenting a project's progress and keeping track of its development.

Practical Skills

At the completion of this unit, students will be able to:

P1. Search, access, and analyse research literature as part of the process of developing solutions to problems.

P2. Understand the importance of analysis, design, documentation, and testing in developing a non-trivial software project.

P3. Write a moderately detailed report explaining methodology, outlining their contributions and the contributions of others, documenting the developed project from developer and user perspectives.

Relationships, Communication and TeamWork

At the completion of this unit, students will be able to:

R1. Understand the role of the client (or user) in the software development process.

R2. Appreciate the importance of written communication in documenting project development.

R3. Understand the importance of assessing time and resource requirements in the successful completion of non-trivial projects.

R4. Appreciate the importance of time and resource management in order to deliver non-trivial projects to deadlines.

Workload

The university standard for a 6-credit point unit is 12 hours of work per week over a semester. Students must be prepared to commit to at least 8 hours of private study per week on this unit, in addition to the contact hours (1 hour per week)

Unit relationships

Prerequisites

Before attempting this unit you must have satisfactorily completed

CSE2304 or CSC2040 or FIT2004 AND CSE2305 or CSC2050 or FIT2001 or FIT2024, or equivalent. Most projects (if not all) will require programming knowledge, although the preferred/required language may vary. Consult the individual project information.

Students underetaking FIT3036 are required to regularly consult the Sakai web page and to follow instructions posted there:

http://sakai.med.monash.edu.au/portal/site/28465a6a-2761-48ff-809e-cfdcbab6e39c/page/99557794-8f94-4666-8029-6782ba

Relationships

FIT3036 is a core unit in the Bachelor of Computer Science degree. There are no units for which this is a prerequisite.

You may not study this unit and CSE3301 in your degree.

Continuous improvement

Monash is committed to ‘Excellence in education’ (Monash Directions 2025 - http://www.monash.edu.au/about/monash-directions/directions.html) and strives for the highest possible quality in teaching and learning.

To monitor how successful we are in providing quality teaching and learning Monash regularly seeks feedback from students, employers and staff. One of the key formal ways students have to provide feedback is through Unit Evaluation Surveys. The University’s Unit Evaluation policy (http://www.policy.monash.edu/policy-bank/academic/education/quality/unit-evaluation-policy.html) requires that every unit offered is evaluated each year. Students are strongly encouraged to complete the surveys as they are an important avenue for students to “have their say”. The feedback is anonymous and provides the Faculty with evidence of aspects that students are satisfied and areas for improvement.

Faculties have the option of administering the Unit Evaluation survey online through the my.monash portal or in class. Lecturers will inform students of the method being used for this unit towards the end of the semester.

Student Evaluations

If you wish to view how previous students rated this unit, please go to http://www.monash.edu.au/unit-evaluation-reports/

Improvements to this unit

SStudent responses to FIT3036 were positive in 2008 and no changes are currently planned.

Unit staff - contact details

Unit leader

Associate Professor John Hurst
Associate Professor
Phone +61 3 990 34102 +61 3 990 55192
Fax +61 3 990 55146

Lecturer(s) :

Mrs Kerri Morgan

Additional communication information

John Hurst, Building 63 Room 63.123

Teaching and learning method

Weekly project group meetings: 1 hour per week

Individual design, coding, testing: 9 hours per week

Tutorial allocation

There are no tutorials in this unit

Communication, participation and feedback

Monash aims to provide a learning environment in which students receive a range of ongoing feedback throughout their studies. You will receive feedback on your work and progress in this unit. This may take the form of group feedback, individual feedback, peer feedback, self-comparison, verbal and written feedback, discussions (on line and in class) as well as more formal feedback related to assignment marks and grades. You are encouraged to draw on a variety of feedback to enhance your learning.

It is essential that you take action immediately if you realise that you have a problem that is affecting your study. Semesters are short, so we can help you best if you let us know as soon as problems arise. Regardless of whether the problem is related directly to your progress in the unit, if it is likely to interfere with your progress you should discuss it with your lecturer or a Community Service counsellor as soon as possible.

Unit Schedule

Week	Topic	Key dates
1	Preliminary reading	
2	Basic plan of attack	
3	(cont.)	
4	(cont.)	
5	Basic parts of some code	
6	(cont.)	
Mid semester break		
7	Basic plan of report	

8	(cont.)	
9	Working prototype of code	
10	Draft report	
11	(cont.)	
12	(cont.)	
13	Completion of report	

Unit Resources

Prescribed text(s) and readings

There are no prescribed text books for this unit. Individual project supervisors may recommend additional reading.

Recommended text(s) and readings

Any textbooks required will be determined by individual project supervisors on a case-by-case basis.

Required software and/or hardware

Projects will normally need access to a computer and programming environment. Individual requirements will be identified by project supervisors

Equipment and consumables required or provided

Students studying off-campus are required to have the minimum system configuration specified by the Faculty as a condition of accepting admission, and regular Internet access.On-campus students, and those studying at supported study locations may use the facilities available in the computing labs.Information about computer use for students is available from the ITS Student Resource Guide in the Monash University Handbook.You will need to allocate up to **n** hours per week for use of a computer, including time for newsgroups/discussion groups.

You need to be aware that the computing resources that may be supplied to you in order to undertake this unit cost real money, and the university has had to impose limits on the use of the internet for all staff and students. A quota system is to be introduced this year, but in the meantime you must make yourself aware of the "Acceptable Use" policies of both the faculty and the university.

These can be accessed on the web at:

- http://www.infotech.monash.edu.au/myfit/students/student_labinfo_rules_netusage.cfm
- http://www.adm.monash.edu.au/unisec/pol/itec12.html

Note that accessing these and other course-related URLs from within Monash is free from within the Monash network, and is not regarded as part of quota.

Study resources

Study resources we will provide for your study are:

During semester 2, 2008, the unit will be supported by Blackboard

Library access

The Monash University Library site contains details about borrowing rights and catalogue searching. To learn more about the library and the various resources available, please go to http://www.lib.monash.edu.au.

The Educational Library and Media Resources (LMR) is also a very resourceful place to visit at http://www.education.monash.edu.au/library/

Monash University Studies Online (MUSO)

All unit and lecture materials are available through MUSO (Monash University Studies Online). Blackboard is the primary application used to deliver your unit resources. Some units will be piloted in Moodle. If your unit is piloted in Moodle, you will see a link from your Blackboard unit to Moodle (http://moodle.monash.edu.au) and can bookmark this link to access directly. In Moodle, from the Faculty of Information Technology category, click on the link for your unit.

You can access MUSO and Blackboard via the portal: http://my.monash.edu.au

Click on the Study and enrolment tab, then Blackboard under the MUSO learning systems.

In order for your Blackboard unit(s) to function correctly, your computer needs to be correctly configured.

For example:

- Blackboard supported browser
- Supported Java runtime environment

For more information, please visit: http://www.monash.edu.au/muso/support/students/downloadables-student.html

You can contact the MUSO Support by phone : (+61 3) 9903 1268

For further contact information including operational hours, please visit: http://www.monash.edu.au/muso/support/students/contact.html

Further information can be obtained from the MUSO support site: http://www.monash.edu.au/muso/support/index.html

Assessment

Unit assessment policy

The unit is assessed on the basis of a completed project report, and work done during the semester (see below). There is no examination in this unit.

Assignment tasks

- ### Assignment Task

  Title : Attendance

  Description :

Meetings are usually held weekly at a time and place convenient to the individual supervisors and each project group. Times & locations will be listed on the third-year notice-board and the online project list (see link above) as soon as they are announced. The first meeting for each group will usually occur in the first week of semester so please check these lists until you have found the time for your first meeting.
Weighting : 10 = min(12x1, 10) marks

Criteria for assessment :

Attendance = 1 mark

Absence = 0 mark

Due date : Weekly, as arranged with supervisor

- **Assignment Task**

  Title : Achievement

  Description :

  This mark will be allocated by the project supervisor, and reflects the outcomes of the project as realised by the student
  Weighting : 50 marks

  Criteria for assessment :

  To be advised by the individual project supervisors, during weekly discussions
  Due date : Friday, 30 May 2008 (end of semester)

  Remarks (optional - leave blank for none) :

  although the due date is 'end of semester', students must make progress during the semester, as determined by their supervisors, and in accordance with the weekly schedule (see below)

- **Assignment Task**

  Title : Report and Presentation

  Description :

  See the weekly schedule
  Weighting : 20 marks

  Criteria for assessment :

  The Project Report provides a complete account of your efforts towards completing the assigned project.

  The contents of the bulk of the report should be organized in a manner which allows the casual reader to quickly identify *what* you were aiming to achieve and *how much* has been achieved. At the same time, the serious reader of the report must be able to easily determine *how* your achievements have been realized.

  One suggested organization for your report is given below; obviously the detailed format and contents is a matter of choice depending upon the nature of the project (especially for non-programming projects) and prior agreement between the student and the supervisor.

Unless told otherwise by the supervisor concerned, the report must be printed on A4 stationery with standard margins. It is anticipated that most reports will be between 8 and 16 pages in length.

2.1. Identification

(1a) Your Name

(1b) Your Student ID

(2) Project Title

(3) Report Date

(4) Supervisor's Name

(5) Access information for the supervisor/examiner to run the project software, such as, machine on which developed, usercode and password.

2.2. General Description

(1) A brief description (in your own words!) of the project task or tasks, including the relationship to other work (by previous students, the supervisor or your peers).

(2) Any theoretical material or reference material relevant to understanding either the task or the solutions to be evaluated and/or employed in the project. It is important to include information on references YOU have consulted and a discussion (where appropriate) of relevant previous work by others (including references to THEIR work). References *must* include full bibliographic data, e.g. Nurd, Fred J. (1983): "Counting the Vertices of a Circle", *Journal of Irreproducible Results*, vol. 7, no 5,.pp 13-12.

(3) A discussion of any assumptions made in developing the solution to the problem, and an evaluation of alternative solutions.

(4) An outline of the method of attack.

2.3. Achievements

A summary, with particular emphasis on limitations of the solution and where the achievements are less than was anticipated (based on the statement of the problem) an explanation of why the goals were not reached.

2.4. Project Documentation

This section should provide a detailed description of the project solution aimed at both users and implementors (modifiers).

2.4.1. User Interface

Essential components of this section include:

1. How to start using the "thing" (program or lump of hardware); configuration and/or initialization parameters, defaults, etc.
2. Complete definition of the user input data and/or command syntax and semantics.
3. Types and meanings of results expected for the various user input options.
4. Permanent data files (e.g. input-output files, database files, libraries); names and uses.

5. Temporary files (e.g. sort scratch); names and uses
6. When the information is available provide estimates of resource consumption (e.g. data file sizes, run-times, report sizes) expressed as mean or typical values and maximum/minimum expected values.
7. In situations where error reporting and recovery is not self-evident, include a step-by-step description of how to recover from automatically detected errors, and a synopsis of error conditions which are not (or cannot be) detected except by the end user.
8. Where the solution provides simple and "bells and whistles" modes of operation, ensure that the user requiring only the simple facilities can readily determine how much of the user interface is relevant.

2.4.2. Implementation

Start with a functional description of what the solution does and a synopsis of how the solution has been structured (e.g. include a call tree or block diagram).

Include a detailed description of all file structures, significant data structures and non-trivial algorithms.

For each substantive module in the solution provide a description of sufficient detail to allow the reader to discover

1. What the module does
2. The interface between the module and the environment in which it must operate (e.g. pin assignments and signals, or parameter definitions and essential global data)
3. The method by which the module implements its stated function
4. The relationship between this module and other modules.

2.4.3. Evidence of Testing

Describe in general terms the tests which have been performed in an attempt to validate the correct operation of the project solution. The emphasis should be on *summarizing* the testing procedures that were employed, rather than including a box of listings and claiming this proves the program was tested.

Where practical, include the transcript of one or more invocations of the solution which demonstrate all the supported features. Such a transcript must be self contained to the extent that another person could reproduce the results using your solution (i.e. include all system-level commands, file assignments, input data, option settings and output data.)

2.5. Concluding Remarks

What has been achieved? Attempt to summarize your own accomplishments, as opposed to the prior achievements of others.

Suggested enhancements to overcome limitations, or to support additional useful facilities.

2.6. References

It is very rarely that a student completes a project without reference to the literature or the prior work of others. Include all books, articles and notes which you used during the course of the project.

2.7. Appendix A

The actual program listings or circuit diagrams which should be well commented and (where possible) cross-referenced.

REMEMBER, REQUIREMENTS VARY DEPENDING ON THE PROJECT AND YOU MUST CONFIRM WITH YOUR SUPERVISOR WHETHER THE ABOVE SCHEME IS APPROPRIATE FOR YOUR PARTICULAR PROJECT.

Due date : 12 June 2009 (end of week 14)

- **Assignment Task**

  Title : Testing

  Description :

  Evidence that the software has been adequately tested
  Weighting : 10 marks

  Criteria for assessment :

  Evidence that the software has been adequately tested
  Due date : 27-28 May 2009 (time to be arranged)

  Remarks (optional - leave blank for none) :

  submit with final report, if not done previously

- **Assignment Task**

  Title : Workbook

  Description :

  A notebook (or computer file) containing weekly entries describing what has been accomplished through the week
  Weighting : 5 marks

  Criteria for assessment :

  At least 10 weekly entries
  Due date : 12 June 2009 (end of week 14)

  Remarks (optional - leave blank for none) :

  submit with final report, if not done previously

- **Assignment Task**

  Title : Demonstration

  Description :

  A demonstration of the software in a working environment
  Weighting : 5 marks

  Criteria for assessment :

  Students will lose 1 mark for each feature not adequately working
  Due date : 22-23 Apr 2009; 27-28 May 2009

Remarks (optional - leave blank for none) :

The two dates are for the two required demonstrations: an Interim Demonstration (Apr); and a Final Demonstration (May). Dates and times for demonstrations will be arranged by each project supervisor

Assignment submission

Assignments should be submitted to the project supervisor on the due dates. A penalty of 10% per day late (including weekends) will be applied to each assignment.

Assignment coversheets

All submitted assignments must include a coversheet, obtained from the "Student assignment coversheets" (http://infotech.monash.edu.au/resources/student/assignments/) page on the faculty website

University and Faculty policy on assessment

Due dates and extensions

The due dates for the submission of assignments are given in the previous section. Please make every effort to submit work by the due dates. It is your responsibility to structure your study program around assignment deadlines, family, work and other commitments. Factors such as normal work pressures, vacations, etc. are seldom regarded as appropriate reasons for granting extensions. Students are advised to NOT assume that granting of an extension is a matter of course.

Late assignment

Assignments received after the due date will be subject to a penalty of 10% per day per assignment (including weekends).

Return dates

Students can expect assignments to be returned within two weeks of the submission date or after receipt, whichever is later.

Assessment for the unit as a whole is in accordance with the provisions of the Monash University Education Policy at http://www.policy.monash.edu/policy-bank/academic/education/assessment/

We will aim to have assignment results made available to you within two weeks after assignment receipt.

Plagiarism, cheating and collusion

Plagiarism and cheating are regarded as very serious offences. In cases where cheating has been confirmed, students have been severely penalised, from losing all marks for an assignment, to facing disciplinary action at the Faculty level. While we would wish that all our students adhere to sound ethical conduct and honesty, I will ask you to acquaint yourself with Student Rights and Responsibilities (http://www.infotech.monash.edu.au/about/committees-groups/facboard/policies/studrights.html) and the Faculty regulations that apply to students detected cheating as these will be applied in all detected cases.

In this University, cheating means seeking to obtain an unfair advantage in any examination or any other written or practical work to be submitted or completed by a student for assessment. It includes the use, or attempted use, of any means to gain an unfair advantage for any assessable work in the unit, where the means is contrary to the instructions for such work.

When you submit an individual assessment item, such as a program, a report, an essay, assignment or other piece of work, under your name you are understood to be stating that this is your own work. If a submission is identical with, or similar to, someone else's work, an assumption of cheating may arise. If you are planning on working with another student, it is acceptable to undertake research together, and discuss problems, but it is not acceptable to jointly develop or share solutions unless this is specified by your lecturer.

Intentionally providing students with your solutions to assignments is classified as "assisting to cheat" and students who do this may be subject to disciplinary action. You should take reasonable care that your solution is not accidentally or deliberately obtained by other students. For example, do not leave copies of your work in progress on the hard drives of shared computers, and do not show your work to other students. If you believe this may have happened, please be sure to contact your lecturer as soon as possible.

Cheating also includes taking into an examination any material contrary to the regulations, including any bilingual dictionary, whether or not with the intention of using it to obtain an advantage.

Plagiarism involves the false representation of another person's ideas, or findings, as your own by either copying material or paraphrasing without citing sources. It is both professional and ethical to reference clearly the ideas and information that you have used from another writer. If the source is not identified, then you have plagiarised work of the other author. Plagiarism is a form of dishonesty that is insulting to the reader and grossly unfair to your student colleagues.

Register of counselling about plagiarism

The university requires faculties to keep a simple and confidential register to record counselling to students about plagiarism (e.g. warnings). The register is accessible to Associate Deans Teaching (or nominees) and, where requested, students concerned have access to their own details in the register. The register is to serve as a record of counselling about the nature of plagiarism, not as a record of allegations; and no provision of appeals in relation to the register is necessary or applicable.

Non-discriminatory language

The Faculty of Information Technology is committed to the use of non-discriminatory language in all forms of communication. Discriminatory language is that which refers in abusive terms to gender, race, age, sexual orientation, citizenship or nationality, ethnic or language background, physical or mental ability, or political or religious views, or which stereotypes groups in an adverse manner. This is not meant to preclude or inhibit legitimate academic debate on any issue; however, the language used in such debate should be non-discriminatory and sensitive to these matters. It is important to avoid the use of discriminatory language in your communications and written work. The most common form of discriminatory language in academic work tends to be in the area of gender inclusiveness. You are, therefore, requested to check for this and to ensure your work and communications are non-discriminatory in all respects.

Students with disabilities

Students with disabilities that may disadvantage them in assessment should seek advice from one of the following before completing assessment tasks and examinations:

- Faculty of Information Technology Student Service staff, and / or
- your Unit Coordinator, or
- Disabilities Liaison Unit

Deferred assessment and special consideration

Deferred assessment (not to be confused with an extension for submission of an assignment) may be granted in cases of extenuating personal circumstances such as serious personal illness or bereavement. Information and forms for Special Consideration and deferred assessment applications are available at http://www.monash.edu.au/exams/special-consideration.html. Contact the Faculty's Student Services staff at your campus for further information and advice.