



ISYS254

Applications Modelling and Development

S1 Day 2017

Dept of Computing

Contents

| | |
|---|----|
| <u>General Information</u> | 2 |
| <u>Learning Outcomes</u> | 3 |
| <u>General Assessment Information</u> | 3 |
| <u>Assessment Tasks</u> | 3 |
| <u>Delivery and Resources</u> | 6 |
| <u>Unit Schedule</u> | 7 |
| <u>Learning and Teaching Activities</u> | 8 |
| <u>Policies and Procedures</u> | 8 |
| <u>Graduate Capabilities</u> | 9 |
| <u>Changes from Previous Offering</u> | 14 |
| <u>Assignment Extension - Policy</u> | 15 |
| <u>Changes from Last Year</u> | 15 |
| <u>Standards and Grading</u> | 15 |

Disclaimer

Macquarie University has taken all reasonable measures to ensure the information in this publication is accurate and up-to-date. However, the information may change or become out-dated as a result of change in University policies, procedures or rules. The University reserves the right to make changes to any information in this publication without notice. Users of this publication are advised to check the website version of this publication [or the relevant faculty or department] before acting on any information in this publication.

General Information

Unit convenor and teaching staff

Convenor and Lecturer

Deborah Richards

deborah.richards@mq.edu.au

Contact via deborah.richards@mq.edu.au

E6A 328

email for an appointment

Lecturer

Charanya Ramakrishnan

charanya.ramakrishnan@mq.edu.au

2nd year lab - during consultation times

TBA

Credit points

3

Prerequisites

ISYS114

Corequisites

Co-badged status

ITEC654

Unit description

This unit is an intermediate unit to deliver a solid foundation in concepts, methods, tools and techniques that organisations use to control the information they use in their day-to-day business, with a particular focus on how computer-based technologies can most effectively contribute to the way business is structured. The unit focuses on the fundamental concepts and models of applications development so that they can understand the key processes related to building functioning applications and appreciate the complexity of applications development. The unit emphasises program development and incorporates the software development life cycle, requirements gathering, designing a solution, and implementing and testing a solution in a programming language.

Important Academic Dates

Information about important academic dates including deadlines for withdrawing from units are available at <https://www.mq.edu.au/study/calendar-of-dates>

Learning Outcomes

On successful completion of this unit, you will be able to:

Practice the key phases of the software development life cycle (SDLC) including requirements engineering, analysis, design, basic development and testing and demonstrate understanding of alternative SDLC lifecycle models.

Demonstrate an understanding of the concepts and tools needed to successfully design and build database-centric application programs using object-oriented and traditional methods and project management techniques.

Demonstrate ability to communicate software requirements and designs clearly and effectively.

General Assessment Information

*If you apply for Disruption to Study for your final examination, you **must** make yourself available for the week of July 24 – 28, 2017. If you are not available at that time, there is no guarantee an additional examination time will be offered. Specific examination dates and times will be determined at a later date.*

Assessment Tasks

| Name | Weighting | Hurdle | Due |
|--|-----------|--------|--------------------------|
| Workshop Participation | 5% | Yes | Weekly |
| Requirements and Modelling | 15% | No | Monday Apr 10, Wk7, 5pm |
| Evaluating Conceptual Design | 5% | No | Tuesday May 2, Wk 8, 5pm |
| App. Dev. Proj - Design | 10% | No | Friday May 12, Wk9, 5pm |
| App. Dev. Proj - Development | 15% | No | Friday June 2, Wk 12 5pm |
| Final Examination | 50% | No | TBA |

Workshop Participation

Due: **Weekly**

Weighting: **5%**

This is a hurdle assessment task (see [assessment policy](#) for more information on hurdle assessment tasks)

Workshops are the key learning activity in this unit and your weekly attendance and active participation is expected. A roll will be taken each week to record your participation. Participation in workshops in Weeks 2- 12 will receive a mark of 0.5 each week (5.5 marks in total is

possible). You **must** participate in at least 5 workshops (thus achieve at least 2.5 out of 5.5) to pass the unit.

Workshops will involve a range of activities, some individual, some in pairs, some in groups. Be prepared to present your ideas.

On successful completion you will be able to:

- Practice the key phases of the software development life cycle (SDLC) including requirements engineering, analysis, design, basic development and testing and demonstrate understanding of alternative SDLC lifecycle models.
- Demonstrate an understanding of the concepts and tools needed to successfully design and build database-centric application programs using object-oriented and traditional methods and project management techniques.

Requirements and Modelling

Due: **Monday Apr 10, Wk7, 5pm**

Weighting: **15%**

This individual assignment will develop your skills in requirements specification, analysis and modelling.

Assignment specification and submission via iLearn.

On successful completion you will be able to:

- Practice the key phases of the software development life cycle (SDLC) including requirements engineering, analysis, design, basic development and testing and demonstrate understanding of alternative SDLC lifecycle models.
- Demonstrate an understanding of the concepts and tools needed to successfully design and build database-centric application programs using object-oriented and traditional methods and project management techniques.
- Demonstrate ability to communicate software requirements and designs clearly and effectively.

Evaluating Conceptual Design

Due: **Tuesday May 2, Wk 8, 5pm**

Weighting: **5%**

You will use a rubric to evaluate the conceptual design of your peers. The designs will be captured in workshops in Week 6.

The details of the rubric and how to access the design will be provided on iLearn.

On successful completion you will be able to:

- Practice the key phases of the software development life cycle (SDLC) including requirements engineering, analysis, design, basic development and testing and demonstrate understanding of alternative SDLC lifecycle models.
- Demonstrate ability to communicate software requirements and designs clearly and effectively.

App. Dev. Proj - Design

Due: **Friday May 12, Wk9, 5pm**

Weighting: **10%**

Application Development Project - Part 1

You will need to form into pairs. Use iLearn to put yourself into a group by the end of week 7. Instructions for doing this will be given in class and in the news forum. Assignment two will have two parts. The first part will assess your design and documentation skills.

Assignment specification will be available on the iLearn site. Submission will be via iLearn.

On successful completion you will be able to:

- Practice the key phases of the software development life cycle (SDLC) including requirements engineering, analysis, design, basic development and testing and demonstrate understanding of alternative SDLC lifecycle models.
- Demonstrate an understanding of the concepts and tools needed to successfully design and build database-centric application programs using object-oriented and traditional methods and project management techniques.
- Demonstrate ability to communicate software requirements and designs clearly and effectively.

App. Dev. Proj - Development

Due: **Friday June 2, Wk 12 5pm**

Weighting: **15%**

Application Development Project - Part 2

The second part of assignment 2 will assess your development and testing skills.

This assignment will be marked in Workshops in Week 13. Both people in the pair must be present to demonstrate their application to receive the mark awarded.

On successful completion you will be able to:

- Practice the key phases of the software development life cycle (SDLC) including requirements engineering, analysis, design, basic development and testing and

demonstrate understanding of alternative SDLC lifecycle models.

- Demonstrate an understanding of the concepts and tools needed to successfully design and build database-centric application programs using object-oriented and traditional methods and project management techniques.
- Demonstrate ability to communicate software requirements and designs clearly and effectively.

Final Examination

Due: **TBA**

Weighting: **50%**

Closed Examination

This closed book exam will test your knowledge of the concepts and ability to apply the learning material for Weeks 1-12.

On successful completion you will be able to:

- Practice the key phases of the software development life cycle (SDLC) including requirements engineering, analysis, design, basic development and testing and demonstrate understanding of alternative SDLC lifecycle models.
- Demonstrate an understanding of the concepts and tools needed to successfully design and build database-centric application programs using object-oriented and traditional methods and project management techniques.
- Demonstrate ability to communicate software requirements and designs clearly and effectively.

Delivery and Resources

CLASSES

Each week you should attend 2 hours of lectures and a workshop (mixed tutorial and practical) Class (starting in week 2). For details of days, times and rooms consult the [timetables webpage](#).

Note that **workshops commence in week 2**.

REQUIRED AND RECOMMENDED TEXTS AND/OR MATERIALS

Textbook

The textbooks for ISYS254 used this semester is:

1. Kenneth E. Kendall and Julie E Kendall., Systems Analysis and Design, Ninth Edition, Pearson, 2012, ISBN13: 978-0-273-78710-5

Note: The book can be found at the co-op bookshop.

UNIT WEBPAGE AND TECHNOLOGY USED AND REQUIRED

Websites

The web page for this unit can be found at: [here](#)

iLecture

Digital recordings of lectures are available. Read instructions [here](#).

Discussion Boards

The unit makes use of discussion boards hosted within [iLearn](#) Please post questions relevant to the unit there. They are monitored by the staff on the unit.

Unit Schedule

| | | | |
|----|--|----------------------------------|--------------|
| 1 | Unit Overview & Understanding and Modelling Organizational Systems | Chp1 & 2 | Richards |
| 2 | Information Gathering + Use Case Diagrams | Chp 4 & 5 and online resources | Richards |
| 3 | O-O Analysis and Design - Class, Sequence Diagrams & State Diagrams | Chp 10 and online resources | Richards |
| 4 | O-O Analysis and Design -Model Consistency, - Package Diagrams | Chp 10 and online resources | Richards |
| 5 | Applications Development and Agile Modeling and Prototyping | Chp 3 and online resources | Ramakrishnan |
| 6 | Applications Development and Database Specification and Modelling | Chp 7 & 8 and online resources | Ramakrishnan |
| 7 | Applications Development and Designing Databases | Chp 13 and Online resources | Ramakrishnan |
| 8 | Applications Development and Human-Computer Interaction | Chp 14 and online resources | Ramakrishnan |
| 9 | Applications Development and Designing Effective Input and Output | Chp 11 & 12 and Online resources | Ramakrishnan |
| 10 | Applications Development and Process Specifications and Structured Decisions | Chp 9 and Online resources | Ramakrishnan |
| 11 | Project Management | Chp 6 | Ramakrishnan |
| 12 | Quality Assurance and Implementation | Chp 15 & 16 | Ramakrishnan |

Learning and Teaching Activities

Lectures

Lectures from staff

Workshops

The workshops provide a combination of tutorial exercises and practicals led by tutors. These workshops are designed to prepare you for all of your assessments by providing individual and group activities that explore the concepts, tools and methods used in requirements gathering, modelling, design and development of applications.

Assignments submission

Submission of assignments related to specific tasks - mix of group and individual assignments.

Exams

A midsemester and final exam will assess individual learning

Policies and Procedures

Macquarie University policies and procedures are accessible from [Policy Central](#). Students should be aware of the following policies in particular with regard to Learning and Teaching:

Academic Honesty Policy http://mq.edu.au/policy/docs/academic_honesty/policy.html

Assessment Policy http://mq.edu.au/policy/docs/assessment/policy_2016.html

Grade Appeal Policy <http://mq.edu.au/policy/docs/gradeappeal/policy.html>

Complaint Management Procedure for Students and Members of the Public http://www.mq.edu.au/policy/docs/complaint_management/procedure.html

Disruption to Studies Policy (in effect until Dec 4th, 2017): http://www.mq.edu.au/policy/docs/disruption_studies/policy.html

Special Consideration Policy (in effect from Dec 4th, 2017): <https://staff.mq.edu.au/work/strategy-planning-and-governance/university-policies-and-procedures/policies/special-consideration>

In addition, a number of other policies can be found in the [Learning and Teaching Category](#) of Policy Central.

Student Code of Conduct

Macquarie University students have a responsibility to be familiar with the Student Code of Conduct: https://students.mq.edu.au/support/student_conduct/

Results

Results shown in *iLearn*, or released directly by your Unit Convenor, are not confirmed as they are subject to final approval by the University. Once approved, final results will be sent to your student email address and will be made available in [eStudent](#). For more information visit [ask.mq.edu.au](#).

Special consideration policy of the Department of Computing:

http://comp.mq.edu.au/undergrad/policies/special_consideration_policy.htm

Student Support

Macquarie University provides a range of support services for students. For details, visit <http://students.mq.edu.au/support/>

Learning Skills

Learning Skills (mq.edu.au/learningskills) provides academic writing resources and study strategies to improve your marks and take control of your study.

- [Workshops](#)
- [StudyWise](#)
- [Academic Integrity Module for Students](#)
- [Ask a Learning Adviser](#)

Student Services and Support

Students with a disability are encouraged to contact the [Disability Service](#) who can provide appropriate help with any issues that arise during their studies.

Student Enquiries

For all student enquiries, visit Student Connect at ask.mq.edu.au

IT Help

For help with University computer systems and technology, visit http://www.mq.edu.au/about_us/offices_and_units/information_technology/help/.

When using the University's IT, you must adhere to the [Acceptable Use of IT Resources Policy](#). The policy applies to all who connect to the MQ network including students.

Graduate Capabilities

Creative and Innovative

Our graduates will also be capable of creative thinking and of creating knowledge. They will be imaginative and open to experience and capable of innovation at work and in the community. We want them to be engaged in applying their critical, creative thinking.

This graduate capability is supported by:

Learning outcome

- Demonstrate an understanding of the concepts and tools needed to successfully design and build database-centric application programs using object-oriented and traditional methods and project management techniques.

Assessment tasks

- Workshop Participation
- Requirements and Modelling
- Evaluating Conceptual Design
- App. Dev. Proj - Design
- App. Dev. Proj - Development
- Final Examination

Capable of Professional and Personal Judgement and Initiative

We want our graduates to have emotional intelligence and sound interpersonal skills and to demonstrate discernment and common sense in their professional and personal judgement. They will exercise initiative as needed. They will be capable of risk assessment, and be able to handle ambiguity and complexity, enabling them to be adaptable in diverse and changing environments.

This graduate capability is supported by:

Learning outcomes

- Demonstrate an understanding of the concepts and tools needed to successfully design and build database-centric application programs using object-oriented and traditional methods and project management techniques.
- Demonstrate ability to communicate software requirements and designs clearly and effectively.

Assessment tasks

- Requirements and Modelling
- App. Dev. Proj - Design
- App. Dev. Proj - Development
- Final Examination

Commitment to Continuous Learning

Our graduates will have enquiring minds and a literate curiosity which will lead them to pursue knowledge for its own sake. They will continue to pursue learning in their careers and as they participate in the world. They will be capable of reflecting on their experiences and relationships with others and the environment, learning from them, and growing - personally, professionally

and socially.

This graduate capability is supported by:

Learning outcome

- Demonstrate ability to communicate software requirements and designs clearly and effectively.

Assessment tasks

- Requirements and Modelling
- App. Dev. Proj - Design
- App. Dev. Proj - Development
- Final Examination

Discipline Specific Knowledge and Skills

Our graduates will take with them the intellectual development, depth and breadth of knowledge, scholarly understanding, and specific subject content in their chosen fields to make them competent and confident in their subject or profession. They will be able to demonstrate, where relevant, professional technical competence and meet professional standards. They will be able to articulate the structure of knowledge of their discipline, be able to adapt discipline-specific knowledge to novel situations, and be able to contribute from their discipline to inter-disciplinary solutions to problems.

This graduate capability is supported by:

Learning outcomes

- Practice the key phases of the software development life cycle (SDLC) including requirements engineering, analysis, design, basic development and testing and demonstrate understanding of alternative SDLC lifecycle models.
- Demonstrate an understanding of the concepts and tools needed to successfully design and build database-centric application programs using object-oriented and traditional methods and project management techniques.

Assessment tasks

- Workshop Participation
- Requirements and Modelling
- App. Dev. Proj - Design
- App. Dev. Proj - Development
- Final Examination

Critical, Analytical and Integrative Thinking

We want our graduates to be capable of reasoning, questioning and analysing, and to integrate and synthesise learning and knowledge from a range of sources and environments; to be able to critique constraints, assumptions and limitations; to be able to think independently and systemically in relation to scholarly activity, in the workplace, and in the world. We want them to have a level of scientific and information technology literacy.

This graduate capability is supported by:

Learning outcomes

- Practice the key phases of the software development life cycle (SDLC) including requirements engineering, analysis, design, basic development and testing and demonstrate understanding of alternative SDLC lifecycle models.
- Demonstrate an understanding of the concepts and tools needed to successfully design and build database-centric application programs using object-oriented and traditional methods and project management techniques.

Assessment tasks

- Workshop Participation
- Requirements and Modelling
- Evaluating Conceptual Design
- App. Dev. Proj - Design
- App. Dev. Proj - Development
- Final Examination

Problem Solving and Research Capability

Our graduates should be capable of researching; of analysing, and interpreting and assessing data and information in various forms; of drawing connections across fields of knowledge; and they should be able to relate their knowledge to complex situations at work or in the world, in order to diagnose and solve problems. We want them to have the confidence to take the initiative in doing so, within an awareness of their own limitations.

This graduate capability is supported by:

Learning outcomes

- Practice the key phases of the software development life cycle (SDLC) including requirements engineering, analysis, design, basic development and testing and demonstrate understanding of alternative SDLC lifecycle models.
- Demonstrate an understanding of the concepts and tools needed to successfully design and build database-centric application programs using object-oriented and traditional

methods and project management techniques.

Assessment tasks

- Workshop Participation
- Requirements and Modelling
- App. Dev. Proj - Design
- App. Dev. Proj - Development
- Final Examination

Effective Communication

We want to develop in our students the ability to communicate and convey their views in forms effective with different audiences. We want our graduates to take with them the capability to read, listen, question, gather and evaluate information resources in a variety of formats, assess, write clearly, speak effectively, and to use visual communication and communication technologies as appropriate.

This graduate capability is supported by:

Learning outcomes

- Demonstrate an understanding of the concepts and tools needed to successfully design and build database-centric application programs using object-oriented and traditional methods and project management techniques.
- Demonstrate ability to communicate software requirements and designs clearly and effectively.

Assessment tasks

- Workshop Participation
- Requirements and Modelling
- App. Dev. Proj - Design
- App. Dev. Proj - Development
- Final Examination

Engaged and Ethical Local and Global citizens

As local citizens our graduates will be aware of indigenous perspectives and of the nation's historical context. They will be engaged with the challenges of contemporary society and with knowledge and ideas. We want our graduates to have respect for diversity, to be open-minded, sensitive to others and inclusive, and to be open to other cultures and perspectives: they should have a level of cultural literacy. Our graduates should be aware of disadvantage and social justice, and be willing to participate to help create a wiser and better society.

This graduate capability is supported by:

Learning outcome

- Demonstrate ability to communicate software requirements and designs clearly and effectively.

Assessment tasks

- Requirements and Modelling
- App. Dev. Proj - Design
- App. Dev. Proj - Development
- Final Examination

Socially and Environmentally Active and Responsible

We want our graduates to be aware of and have respect for self and others; to be able to work with others as a leader and a team player; to have a sense of connectedness with others and country; and to have a sense of mutual obligation. Our graduates should be informed and active participants in moving society towards sustainability.

This graduate capability is supported by:

Learning outcome

- Demonstrate ability to communicate software requirements and designs clearly and effectively.

Assessment tasks

- Requirements and Modelling
- App. Dev. Proj - Design
- App. Dev. Proj - Development
- Final Examination

Changes from Previous Offering

The weekly structure has changed to focus on requirements and UML modelling in the first half and design, project management and quality in the second half.

Students will work on examples and problems in mixed classes to prepare them for the assignments which will target similar (but not the same) problem.

Students will learn how OO analysis and design relates to OO software development and also create non-OO analysis and designs and associated code.

Assignment 2 will be group based and Assignment 1 will be an individual project. Both assignment will be delivered in two phases.

Assignment Extension - Policy

No extensions will be granted. Late assignments will be accepted up to 72 hours after the submission deadline. There will be a deduction of 20%* of the total available marks made from the total awarded mark for each 24 hour period or part thereof that the submission is late (for example, 25 hours late in submission – 40% penalty). This penalty does not apply for cases in which an application for special consideration is made and approved.

Changes from Last Year

Standards and Grading

In order to pass the unit, you must obtain a total mark of 50% or higher in the unit. The final mark will be the summation of the marks you have received for assessments. Participation in workshops is a hurdle requirement. You are required to participate in at least 5 weekly workshops (combined practical and tutorial) to be eligible to sit the final exam.

ISYS254 will be graded according to the following general descriptions of the letter grades as specified by Macquarie University.

- High Distinction (HD, 85-100): provides consistent evidence of deep and critical understanding in relation to the learning outcomes. There is substantial originality and insight in identifying, generating and communicating competing arguments, perspectives or problem solving approaches; critical evaluation of problems, their solutions and their implications; creativity in application as appropriate to the discipline.
- Distinction (D, 75-84): provides evidence of integration and evaluation of critical ideas, principles and theories, distinctive insight and ability in applying relevant skills and concepts in relation to learning outcomes. There is demonstration of frequent originality in defining and analysing issues or problems and providing solutions; and the use of means of communication appropriate to the discipline and the audience.
- Credit (Cr, 65-74): provides evidence of learning that goes beyond replication of content knowledge or skills relevant to the learning outcomes. There is demonstration of substantial understanding of fundamental concepts in the field of study and the ability to apply these concepts in a variety of contexts; convincing argumentation with appropriate coherent justification; communication of ideas fluently and clearly in terms of the conventions of the discipline..
- Pass (P, 50-64): provides sufficient evidence of the achievement of learning outcomes. There is demonstration of understanding and application of fundamental concepts of the field of study; routine argumentation with acceptable justification; communication of information and ideas adequately in terms of the conventions of the discipline. The learning attainment is considered satisfactory or adequate or competent or capable in relation to the specified outcomes.

- Fail (F, 0-49): does not provide evidence of attainment of learning outcomes. There is missing or partial or superficial or faulty understanding and application of the fundamental concepts in the field of study; missing, undeveloped, inappropriate or confusing argumentation; incomplete, confusing or lacking communication of ideas in ways that give little attention to the conventions of the discipline.

| L.O 1 | Developing | Functional(P) | Proficient (Cr-D) | Advanced (D-HD) |
|---|--|--|--|---|
| Practice the key phases of the software development life cycle (SDLC) including requirements engineering, analysis, design, basic development and testing and demonstrate understanding of alternative SDLC lifecycle models. | Basic knowledge of the names of the phases of the traditional system development life cycle but lack of understanding of alternative life cycles, how the phases relate, artefacts produced and models used in each phase or how the life cycle is used to turn a real world problem into a software solution. | Awareness of the phases of the traditional system development life cycle some understanding of alternative life cycles, how the phases relate, artefacts produced and models used in each phase or how the life cycle is used to turn a real world problem into a software solution. | Understanding of traditional and alternative life cycles, how the phases relate, artefacts produced and models used in each phase or how the life cycle is used to turn a real world problem into a software solution. Awareness of issues such as quality, project management, methodologies. | Deep understanding of traditional and alternative life cycles, how the phases relate, artefacts produced and models used in each phase or how the life cycle is used to turn a real world problem into a software solution. Appreciation of a range of issues such as quality, project management, design tradeoffs and choices, methodologies and how they impact on the life cycle, the project, the team and software developed. |
| L.O. 2 | Developing | Functional | Proficient | Advanced |
| Demonstrate an understanding of the concepts and tools needed to successfully design and build database-centric application programs using object-oriented and traditional methods and project management techniques. | Ability to use a CASE tool to create models but with limited understanding of how the models connect and how the tool can be effectively used to ensure model consistency. | Ability to use a CASE tool to create models, understanding of how the models connect and some understanding of how the tool can be effectively used to ensure model consistency. | Ability to use a CASE tool to create models, understanding of how the models connect and how the tool can be effectively used to ensure model consistency. | Ability to use a CASE tool to create models, understanding of how the models connect and how the tool can be effectively and efficiently used to ensure model consistency. Familiarity with extended features of the tools and the role that these features can play in supporting the development of software. |
| L.O.3 | Developing | Functional | Proficient | Advanced |
| Demonstrate ability to communicate software requirements and designs clearly and effectively. | Aware of documentation standards and notations but not able to apply them to the appropriate situation. | Able to follow a standard (e.g. IEEE standard for SRS) and appropriately use modelling notations. Able to document and read documentation concerning a software application. | Able to communicate at a functional level and also able to verify and validate documentation produced by themselves and others. | Able to communicate at a proficient level and also able to demonstrate a deep understanding of the role and relationship of various documents, activities, processes and roles which make up software development teams, projects, processes and products. |