COMP7010

Advanced Topics in Theory and Practice of Software

Session 1, Weekday attendance, North Ryde 2020

Department of Computing

Contents

General Information	2
Learning Outcomes	2
General Assessment Information	3
Assessment Tasks	3
Delivery and Resources	4
Unit Schedule	5
Policies and Procedures	6
Changes from Previous Offering	8
Changes since First Published	8

General Information

Unit convenor and teaching staff
Convenor, Lecturer
Anthony Sloane
anthony.sloane@mq.edu.au
Contact via Email
4RPD267
By appointment

Credit points
10

Prerequisites
Admission to MRes

Corequisites

Co-badged status

Unit description
This unit introduces the formal study of software systems. It is intended to provide a general basis for further study or research in software-focused areas of Computer Science such as Programming Languages and Formal Methods. The unit is organised around two main themes: a) the meaning of languages and programs, and b) techniques for verifying that languages and programs have desired properties. The practical work in the unit includes implementation of formal language semantics and development of verification proofs.

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:

ULO1: Understand, specify, analyse and evaluate the meaning of programming languages and programs using mathematical techniques.

ULO2: Apply a practical tool or system to specify and prove formal properties of programming languages and programs.

ULO3: Research a topic in advanced computer science and report findings in written and oral form.

Assessment Tasks

Coronavirus (COVID-19) Update

Assessment details are no longer provided here as a result of changes due to the Coronavirus (COVID-19) pandemic.

Students should consult iLearn for revised unit information.

Find out more about the Coronavirus (COVID-19) and potential impacts on staff and students

General Assessment Information

COMP7010 will be assessed and graded according to the University assessment and grading policies.

Submission Deadlines

Assessment deadlines are strict, unless an application for special consideration is received (preferably in advance) accompanied by appropriate documentary evidence. Late submissions will be penalised at the rate of 20% of the full marks for the assessment per day or part thereof late.

Standards

The following general standards of achievement will be used to assess each of the assessment tasks with respect to the letter grades.

Pass: Has a basic undestanding of language and program semantics as discussed in class. Can use a verification tool or proof system to specify and prove simple language and program properties similar to those discussed in class. Can perform a basic research investigation in the area and present the results of that research in rudimentary written and oral forms.

Credit: As for Pass plus: Is able to apply the techniques we have discussed to specify and prove new properties that are not direct analogues of ones discussed in class. Shows more than basic insights into the results of a research investgiation and is able to communicate those insights.

Distinction/High Distinction: As for Credit plus: Is able to generalise from the language and program properties discussed explicity to new ones for problem domains not explicitly discussed in class and can apply verification tools and systems to proofs about them. Can critically evaluate the limits of the techniques and tools we have discussed.

Assessment Process

These assessment standards will be used to give a numeric mark out of 100 to each assessment submission during marking. The mark will correspond to a letter grade for that task according to the University guidelines. The final mark for the unit will be calculated by combining the marks for all assessment tasks according to the percentage weightings shown in the assessment summary.

Special Consideration

If you receive special consideration for the final exam, a supplementary exam will be scheduled in the interval between the regular exam period and the start of the next session. By making a special consideration application for the final exam you are declaring yourself available for a resit during the supplementary examination period and will not be eligible for a second special consideration approval based on pre-existing commitments.Please ensure you are familiar with the policy prior to submitting an application. Approved applicants will receive an individual notification at least one week prior to the exam with the exact date and time of their supplementary examination.

Delivery and Resources

Coronavirus (COVID-19) Update

Any references to on-campus delivery below may no longer be relevant due to COVID-19.

Please check here for updated delivery information: https://ask.mq.edu.au/account/pub/display/unit_status

CLASSES

Each week of COMP7010 has two to three hours of face-to-face class. Classes will be a mixture of lecture-style presentation, discussion and practical demonstration.

REQUIRED AND RECOMMENDED TEXTS AND/OR MATERIALS

COMP7010 will follow the book "Software Foundations" by Benjamin Pierce et al. The book consists of annotated programs, proofs and exercises. It can be read in HTML form or the full distribution can be downloaded for formatting as PDF. This book is updated from time to time.

We will use Version 4.0 (May 2016) of the book. Note that more recent versions of the book have been significantly reorganised, split into multiple books, so a current version will not work for this unit.

Students should read the relevant sections of the book and attempt the basic exercises on their own. Class time will be devoted to the main ideas of each chapter and working through a number of exercises. Some basic and more advanced exercises will be set as homework exercises.

UNIT WEBPAGE AND TECHNOLOGY USED AND REQUIRED

COMP7010 uses iLearn for delivery of class materials, discussion boards, online selftests, submission of assessment tasks and access to marks and comments. Students should check the iLearn site regularly for unit updates.

Questions regarding the content of this unit should be posted to the appropriate discussion board on iLearn. In particular, any questions which are of interest to all students in this unit should be posted to one of these discussion boards, so that everyone can benefit from the answers.

Coq Proof Assistant

The practical work in this unit involves functional programming and proof construction using the Coq proof assistant (http://coq.inria.fr). Ideally, students should install Coq on a laptop and bring it to class so they can follow along with in-class exercises.

We will use Coq version 8.11.0.

Coq is usually used via a simple IDE platform of which the easiest one is the CoqIDE that is available as part of the Coq distribution. Students who are familiar with Emacs may want to look at the Proof General interface which is similar to CoqIDE but embedded in Emacs. Another option is VsCoq which is a Coq extension for the Visual Studio Code editor.

Unit Schedule

Coronavirus (COVID-19) Update

The unit schedule/topics and any references to on-campus delivery below may no longer be relevant due to COVID-19. Please consult iLearn for latest details, and check here for updated delivery information: https://ask.mq.edu.au/account/pub/display/unit_status

The unit introduces the formal study of software systems. It is intended to provide a general basis for further study or research in software-focused areas of Computer Science such as Programming Languages and Formal Methods.

The unit is organised around two main themes:

a) The meaning of programs. To study software systems it is necessary to have a proper understanding of the programming languages in which those systems are written. Formal semantic descriptions of languages assign mathematical meanings to programs. Typical kinds of meaning that will be studied include types that specify the operations that programs will perform, and operational aspects that capture how a program behaves as it executes.

b) Techniques for verifying that languages and programs have desired properties. We will see how to analyse a language semantics to prove that all programs written in that language have desirable properties. E.g., a desirable property of a type system is type safety: that an executing program cannot execute an illegal operation. We will also study the properties of particular programs. E.g., it is often desirable to be able to prove that a program produces a desired result.

The practical work in the unit will include implementation of formal language semantics and development of verification proofs. We emphasise the use of frameworks and tools to assist with both of these activities. Examples include the use of software language engineering tools and libraries to assist with language implementation, and the use of proof assistants, program verification systems or model checkers to help us specify properties and to find proofs.

Students entering this unit should have reasonable programming experience and should have studied discrete mathematics. Relevant Macquarie University units for programming maturity are

COMP2000 Object-Oriented Programming Practices, COMP2010 Algorithms and Data Structures, and any 3000-level unit that applies programming skills to particular problem domains (e.g., COMP3170 Computer Graphics or COMP3010 Algorithm Theory and Design). MATH1007 Discrete Mathematics I and MATH2907 Discrete Mathematics II provide good mathematical background. Previous courses in areas such as programming language concepts or implementation, such as COMP3000 Programming Languages, will be helpful, but are not required.

The class material will be structured according to the following schedule. The rightmost column refers to the relevant chapters of the "Software Foundations" required text.

Week	Topic	Chapter(s)
	Introduction, Functional Programming	Preface, Basics
	Proof by Induction, Structured Data	Induction, Lists
	Polymorphism and Higher-Order Functions	Poly
	More Basic Tactics	Tactics
	Logical Reasoning	Logic
	Inductive Propositions	IndProp
	Maps, Imperative Programming Language	Maps, Imp
	Program Equivalence	Equiv
	Hoare Logic for Imperative Programs	Hoare
	Smallstep Operational Semantics	Smallstep
	Simple-typed Lambda Calculus	Stlc, StlcProp
	Student Presentations	

Policies and Procedures

Macquarie University policies and procedures are accessible from Policy Central (https://staff.mq.edu.au/work/strategy-planning-and-governance/university-policies-and-procedures/policy-central). Students should be aware of the following policies in particular with regard to Learning and Teaching:

- Academic Appeals Policy
- Academic Integrity Policy
- Academic Progression Policy
- Assessment Policy

- Fitness to Practice Procedure
- Grade Appeal Policy
- Complaint Management Procedure for Students and Members of the Public
- Special Consideration Policy *(**Note:** The Special Consideration Policy is effective from 4 December 2017 and replaces the Disruption to Studies Policy.)*

Students seeking more policy resources can visit the Student Policy Gateway (https://students.mq.edu.au/support/study/student-policy-gateway). It is your one-stop-shop for the key policies you need to know about throughout your undergraduate student journey.

If you would like to see all the policies relevant to Learning and Teaching visit Policy Central (https://staff.mq.edu.au/work/strategy-planning-and-governance/university-policies-and-procedures/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/study/getting-started/student-conduct

Results

Results published on platform other than eStudent, (eg. iLearn, Coursera etc.) 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 or if you are a Global MBA student contact globalmba.support@mq.edu.au

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 help you improve your marks and take control of your study.

- Getting help with your assignment
- Workshops
- StudyWise
- Academic Integrity Module

The Library provides online and face to face support to help you find and use relevant information resources.

- Subject and Research Guides
- Ask a Librarian

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

If you are a Global MBA student contact globalmba.support@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.

Changes from Previous Offering

There are no significant changes this year. We will use the latest release of Coq but remain with the previously used version of the textbook.

Changes since First Published

Date	Description
05/02/2020	Unit schedule: fix that Week 8 Monday is not a holiday; put Week 13 student presentations into schedule table. Change unit codes of possible useful previous units to new codes. Move assessment standards to assessment information section.
17/01/2020	Put assessments into a more sensible order.