COMP7010
Advanced Topics in Theory and Practice of Software
Session 1, Weekday attendance, North Ryde 2020

Department of Computing

Coronavirus (COVID-19) Update
Due to the Coronavirus (COVID-19) pandemic, any references to assessment tasks and on-campus delivery may no longer be up-to-date on this page.
Students should consult iLearn for revised unit information.
Find out more about the Coronavirus (COVID-19) and potential impacts staff and students

Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>General Information</td>
<td>2</td>
</tr>
<tr>
<td>Learning Outcomes</td>
<td>2</td>
</tr>
<tr>
<td>General Assessment Information</td>
<td>0</td>
</tr>
<tr>
<td>Assessment Tasks</td>
<td>3</td>
</tr>
<tr>
<td>Delivery and Resources</td>
<td>3</td>
</tr>
<tr>
<td>Unit Schedule</td>
<td>4</td>
</tr>
<tr>
<td>Policies and Procedures</td>
<td>5</td>
</tr>
<tr>
<td>Changes from Previous Offering</td>
<td>7</td>
</tr>
<tr>
<td>Changes since First Published</td>
<td>7</td>
</tr>
</tbody>
</table>

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, 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://students.mq.edu.au/important-dates

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

UL01: Understand, specify, analyse and evaluate the meaning of programming languages and programs using mathematical techniques.

UL02: Apply a practical tool or system to specify and prove formal properties of programming languages and programs.

UL03: 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 staff and students

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.

<table>
<thead>
<tr>
<th>Week</th>
<th>Topic</th>
<th>Chapter(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Introduction, Functional Programming</td>
<td>Preface, Basics</td>
</tr>
<tr>
<td>2</td>
<td>Proof by Induction, Structured Data</td>
<td>Induction, Lists</td>
</tr>
<tr>
<td>3</td>
<td>Polymorphism and Higher-Order Functions</td>
<td>Poly</td>
</tr>
<tr>
<td>4</td>
<td>More Basic Tactics</td>
<td>Tactics</td>
</tr>
<tr>
<td>5</td>
<td>Logical Reasoning</td>
<td>Logic</td>
</tr>
<tr>
<td>6</td>
<td>Inductive Propositions</td>
<td>IndProp</td>
</tr>
<tr>
<td>7</td>
<td>Maps, Imperative Programming Language</td>
<td>Maps, Imp</td>
</tr>
<tr>
<td>8</td>
<td>Program Equivalence</td>
<td>Equiv</td>
</tr>
<tr>
<td>9</td>
<td>Hoare Logic for Imperative Programs</td>
<td>Hoare</td>
</tr>
<tr>
<td>10</td>
<td>Smallstep Operational Semantics</td>
<td>Smallstep</td>
</tr>
<tr>
<td>11-12</td>
<td>Simple-typed Lambda Calculus</td>
<td>Stlc, StlcProp</td>
</tr>
<tr>
<td>13</td>
<td>Student Presentations</td>
<td></td>
</tr>
</tbody>
</table>

**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**
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 improve your marks and take control of your study.

- Workshops
- StudyWise
- Academic Integrity Module for Students
- Ask a Learning Adviser

**Student Enquiry Service**

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
Equity 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.

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

<table>
<thead>
<tr>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>05/02/2020</td>
<td>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.</td>
</tr>
<tr>
<td>17/01/2020</td>
<td>Put assessments into a more sensible order.</td>
</tr>
</tbody>
</table>