COMP3000

Programming Languages

Session 2, In person-scheduled-weekday, North Ryde 2023

School of Computing

Contents

General Information

Unit convenor and teaching staff
Lecturer, Convener
Matthew Roberts
matthew.roberts@mq.edu.au
TBA

Lecturer
Kym Haines
kym.haines@mq.edu.au
TBA

Credit points
10

Prerequisites
130cp at 1000 level or above including COMP2010 or COMP225 or COMP2000 or COMP229

Corequisites

Co-badged status

Unit description
Formal languages play a central role in modern software development. Programming languages such as Java and C++ allow developers to express their algorithms and data structures. Compilers and interpreters transform programs into running software. Data languages such as XML and JSON are widely used to transfer information between systems. This unit studies software languages by looking at how they are used in software development. Students will study how to formally understand the syntax, semantics and translation of software languages. Practical exercises involve writing software language processors of various kinds such as simple compilers or data transformation tools.

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: evaluate the role that languages play in software development and describe a spectrum of software languages that are in current use

ULO2: express properties of software languages using formal notations

ULO3: translate formal notations of software language properties into implementations of language processors

ULO4: defend the correct operation of a language processor by construction and use of appropriate test cases

General Assessment Information

There will be one exam question per week in class in most weeks, excluding week one. Usually the question will relate to some aspect of that class' content or possibly material from previous lectures or classes.

The final exam will offer a second chance to do at most six of the weekly exam questions of your choosing (i.e. choosing the weeks, not the actual questions), covering the same subject area(s) of that week.

Late Assessment Submission Penalty

Unless a Special Consideration request has been submitted and approved, a 5% penalty (of the total possible mark of the task) will be applied for each day a written report or presentation assessment is not submitted, up until the 7th day (including weekends). After the 7th day, a grade of ‘0’ will be awarded even if the assessment is submitted. The submission time for all uploaded assessments is 11:55 pm. A 1-hour grace period will be provided to students who experience a technical concern. For any late submission of time-sensitive tasks, such as scheduled tests/exams, performance assessments/presentations, and/or scheduled practical assessments/labs, please apply for Special Consideration.

Assessments where Late Submissions will be accepted

In this unit, late submissions will be accepted as follows:

- Programming Assessments: YES
- Weekly SGTA Tasks: NO, unless special consideration is granted

Requirements to Pass

To pass this unit you need to accumulate more than 49 marks overall from the various weighted assessments.

Special Consideration

The Special Consideration Policy aims to support students who have been impacted by short-term circumstances or events that are serious, unavoidable and significantly disruptive, and which may affect their performance in assessment. If you experience circumstances or events that affect your ability to complete the assessments in this unit on time, please inform the convenor and submit a Special Consideration request through ask.mq.edu.au.

Assessment Tasks

Name	Weighting	Hurdle	Due
Translation	15%	No	Week 12
Syntax Analysis	15%	No	Mid-Semester Break
Scala	10%	No	Week 5
Examinations	60%	No	One question per week in class

Translation

Assessment Type [1]: Programming Task
Indicative Time on Task [2]: 15 hours
Due: **Week 12**
Weighting: **15%**

The third assignment focuses on translating a language into some other form, such as another structured language (e.g., translating a programming language into a lower-level form such as bytecode or assembly language).

On successful completion you will be able to:

- express properties of software languages using formal notations
- translate formal notations of software language properties into implementations of language processors
- defend the correct operation of a language processor by construction and use of appropriate test cases

Syntax Analysis

Assessment Type [1]: Programming Task
Indicative Time on Task [2]: 15 hours
Due: **Mid-Semester Break**
Weighting: **15%**

The second assignment focuses on processing the syntax of a language to obtain a representation that the rest of the implementation can use.

On successful completion you will be able to:

- express properties of software languages using formal notations
- translate formal notations of software language properties into implementations of language processors
- defend the correct operation of a language processor by construction and use of appropriate test cases

Scala

Assessment Type [1]: Programming Task
Indicative Time on Task [2]: 15 hours
Due: **Week 5**
Weighting: **10%**

The first assignment focuses on using Scala (particularly its functional features) to develop a small-medium-sized program. The aim is to consolidate and assess Scala programming skills in preparation for the other two assignments.

On successful completion you will be able to:

- express properties of software languages using formal notations
- translate formal notations of software language properties into implementations of language processors
- defend the correct operation of a language processor by construction and use of appropriate test cases

Examinations

Assessment Type [1]: Examination
Indicative Time on Task [2]: 30 hours
Due: **One question per week in class**
Weighting: **60%**

A number of exams spread through the semester.

On successful completion you will be able to:

- evaluate the role that languages play in software development and describe a spectrum

of software languages that are in current use
- express properties of software languages using formal notations

---

[1] If you need help with your assignment, please contact:

- the academic teaching staff in your unit for guidance in understanding or completing this type of assessment
- the Writing Centre for academic skills support.

[2] Indicative time-on-task is an estimate of the time required for completion of the assessment task and is subject to individual variation

Delivery and Resources

CLASSES

Each week of COMP3000 has two hours of lecture and a two-hour class. The classes will require a mixture of tutorial-style and practical work. Classes start in Week 1.

REQUIRED AND RECOMMENDED TEXTS AND/OR MATERIALS

There is no required text. We will provide notes or references to freely available materials where relevant.

The free book Creative Scala (https://www.creativescala.org) is a clear introduction to functional programming in Scala.

Students may find it useful to consult one of the many books that are available on the programming languages topic. The following books are among those that are available in the Macquarie University Library:

- Programming Language Pragmatics. Scott.
- Principles of programming languages: design, evaluation, and implementation. MacLennan.
- Programming languages: design and implementation. Pratt and Zelkowitz.
- Concepts of programming languages. Sebesta.
- Programming languages: concepts and constructs. Sethi.
- Introduction to compiler construction. Waite and Carter.
- Compilers: principles, techniques and tools. Aho, Sethi, and Ullman.
- Modern compiler implementation in Java. Appel.

UNIT WEBPAGE AND TECHNOLOGY USED AND REQUIRED

COMP3000 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, its tutorials or practicals 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.

The practical work in this unit mostly involves programming in the Scala language (http://www.scala-lang.org) which will give students experience with modern programming language features that we expect to see in mainstream languages in the future.

We will also use the Kiama language processing library (https://bitbucket.org/inkytonik/kiama) that is being developed by our Programming Languages and Verification Research Group. Kiama provides high-level facilities for writing processors such as compilers in Scala and makes it possible for students to implement of a language from scratch within the semester.

Instructions will be provided on how to use Scala and Kiama on the laboratory machines and how to download it for use on your own machines.

Unit Schedule

Topics covered include:

- Scala
- functional programming
- names
- types
- syntax analysis
- semantic analysis
- transformation
- compilation
- subroutines
- control abstraction
- data abstraction
- object-oriented programming
- language runtimes
- interpretation

Policies and Procedures

Macquarie University policies and procedures are accessible from Policy Central (https://policies.mq.edu.au). 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
- Assessment Procedure
- Complaints Resolution Procedure for Students and Members of the Public
- Special Consideration Policy

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

To find other policies relating to Teaching and Learning, visit Policy Central (https://policies.mq.edu.au) and use the search tool.

Student Code of Conduct

Macquarie University students have a responsibility to be familiar with the Student Code of Conduct: https://students.mq.edu.au/admin/other-resources/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

Academic Integrity

At Macquarie, we believe academic integrity – honesty, respect, trust, responsibility, fairness and courage – is at the core of learning, teaching and research. We recognise that meeting the expectations required to complete your assessments can be challenging. So, we offer you a range of resources and services to help you reach your potential, including free online writing and maths support, academic skills development and wellbeing consultations.

Student Support

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

The Writing Centre

The Writing Centre provides resources to develop your English language proficiency, academic writing, and communication skills.

- Workshops
- Chat with a WriteWISE peer writing leader
- Access StudyWISE
- Upload an assignment to Studiosity

- Complete the 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

Macquarie University offers a range of Student Support Services including:

- IT Support
- Accessibility and disability support with study
- Mental health support
- Safety support to respond to bullying, harassment, sexual harassment and sexual assault
- Social support including information about finances, tenancy and legal issues
- Student Advocacy provides independent advice on MQ policies, procedures, and processes

Student Enquiries

Got a question? Ask us via AskMQ, or contact Service Connect.

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

none

Changes since First Published

Date	Description
04/10/2023	"Tutorials" removed and replaced with SGTA ("Small Group Teaching Activities)

Unit information based on version 2023.02 of the Handbook