



COMP3000

Programming Languages

Session 2, Special circumstance 2020

Department of Computing

Contents

<u>General Information</u>	2
<u>Learning Outcomes</u>	2
<u>General Assessment Information</u>	3
<u>Assessment Tasks</u>	3
<u>Delivery and Resources</u>	6
<u>Unit Schedule</u>	7
<u>Policies and Procedures</u>	8

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.

Notice

As part of [Phase 3 of our return to campus plan](#), most units will now run tutorials, seminars and other small group learning activities on campus for the second half-year, while keeping an online version available for those students unable to return or those who choose to continue their studies online.

To check the availability of face-to-face and online activities for your unit, please go to [timetable viewer](#). To check detailed information on unit assessments visit your unit's iLearn space or consult your unit convenor.

General Information

Unit convenor and teaching staff

Kym Haines

kym.haines@mq.edu.au

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

Late Submission

No extensions will be granted without an approved application for Special Consideration. 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 for an assignment worth 10 marks – 40% penalty or 4 marks deducted from the total. No submission will be accepted after solutions have been posted.

Assessment Tasks

Name	Weighting	Hurdle	Due
Scala	10%	No	Week 6
Syntax Analysis	15%	No	Week 9
Translation	15%	No	Week 12
Examination One	20%	No	Week 7
Examination Two	20%	No	Week 10
Examination Three	20%	No	Week 13

Scala

Assessment Type ¹: Programming Task

Indicative Time on Task ²: 15 hours

Due: **Week 6**

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

Syntax Analysis

Assessment Type ¹: Programming Task

Indicative Time on Task ²: 15 hours

Due: **Week 9**

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

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

Examination One

Assessment Type ¹: Examination

Indicative Time on Task ²: 10 hours

Due: **Week 7**

Weighting: **20%**

This examination will assess the material from Weeks 1-4 of 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

Examination Two

Assessment Type ¹: Examination

Indicative Time on Task ²: 10 hours

Due: **Week 10**

Weighting: **20%**

This examination will assess the material from Weeks 5-8 of 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

Examination Three

Assessment Type ¹: Examination

Indicative Time on Task ²: 10 hours

Due: **Week 13**

Weighting: **20%**

This examination will assess the material from Weeks 9-12 of 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 three 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 and we will use the Doodle graphics library described in that book to illustrate Scala programming principles.

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 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

Week	Topic	Assignment Due	Exam
1	Introduction: Unit and Scala Basics		
2-3	Functional Programming in Scala		
4	Syntax		
5	Names		
6	Semantic analysis	One	
7	Types		One
	Lecture Recess - Two Weeks		
8	Semantic analysis: an example		
9	Transformation; compilation	Two	
10	Subroutines and control abstraction		Two

Week	Topic	Assignment Due	Exam
11	Data abstraction and object-oriented programming		
12	Language runtimes; interpretation	Three	
13	Review, Exam Discussion		Three
Exam Period			One, Two, Three

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) (<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) (<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) (<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.