COMP3000
Programming Languages
Session 2, Special circumstances 2021

Department of Computing

Contents

General Information 2
Learning Outcomes 2
General Assessment Information 3
Assessment Tasks 3
Delivery and Resources 6
Unit Schedule 7
Policies and Procedures 7

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
Some on-campus classes have moved online for the first two weeks of Session, before returning to campus in Week 3. If you are studying a unit outside of the primary Session 2 timetable, please contact your teaching staff team for further details.

Some classes/teaching activities cannot be moved online and must be taught on campus. To find out if you are enrolled in one of these classes/teaching activities, you can check to see if your unit is on the list of units with mandatory on-campus classes/teaching activities.

Your Unit Convenor will provide more information via an iLearn announcement when your iLearn unit becomes available.
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

Co-requisites

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://students.mq.edu.au/important-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. (Ignore the week ranges for exam content.)

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Weighting</th>
<th>Hurdle</th>
<th>Due</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scala</td>
<td>10%</td>
<td>No</td>
<td>week 6</td>
</tr>
<tr>
<td>Syntax Analysis</td>
<td>15%</td>
<td>No</td>
<td>week 8</td>
</tr>
<tr>
<td>Translation</td>
<td>15%</td>
<td>No</td>
<td>week 13</td>
</tr>
<tr>
<td>Examination One</td>
<td>20%</td>
<td>No</td>
<td>one question per week</td>
</tr>
<tr>
<td>Examination Two</td>
<td>20%</td>
<td>No</td>
<td>one question per week</td>
</tr>
<tr>
<td>Examination Three</td>
<td>20%</td>
<td>No</td>
<td>one question per week</td>
</tr>
</tbody>
</table>

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
Syntax Analysis

Assessment Type 1: Programming Task
Indicative Time on Task 2: 15 hours
Due: week 8
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 13
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: one question per week
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: one question per week
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: one question per week
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 Learning Skills Unit 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](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.
- 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://staff.mq.edu.au/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 (http://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/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

**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
**Student Enquiry Service**

For all student enquiries, visit Student Connect at [ask.mq.edu.au](http://ask.mq.edu.au)

If you are a Global MBA student contact [globalmba.support@mq.edu.au](mailto:globalmba.support@mq.edu.au)

**Equity Support**

Students with a disability are encouraged to contact the [Disability Service](mailto:disability.service@mq.edu.au) 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/](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](https://www.mq.edu.au/about_us/offices_and_units/information_technology/acceptable_use_policy/).

The policy applies to all who connect to the MQ network including students.