

## **UNIT GUIDE**

# **COMP115 Introduction to Computer Science**

First Half 2011

**Credit points:** Three

**Pre-requisites:** None

### **Teaching Staff**

*Lecturer and Convenor (Weeks 1-4, 8-9, 13)*

Associate Professor Tony Sloane, Anthony.Sloane@mq.edu.au, E6A315, 98509582

Consultation times: Immediately after a lecture; on Moodle; Thursday, 5-6pm in E6A315; or by appointment

*Lecturer and Convenor (Weeks 5-7, 10-13)*

Associate Professor Dominic Verity, domv@science.mq.edu.au, E6A325, 98509522

Consultation times: Immediately after a lecture; on Moodle; Thursday, 5-6pm in E6A325; or by appointment

### **Co-badging**

COMP115 is not co-badged with any other unit code.

### **Unit description**

COMP115 is the main introductory Computer Science unit, which provides a practical introduction to basic computing and programming concepts. Students will gain an understanding of, and practical experience in, computer programming; practical experience in implementing informal prose descriptions of problem solutions using an imperative language; an understanding of, and practical experience in, designing, coding, testing and 'debugging' simple algorithms; and an understanding of the principle of incremental development.

Other topics include the concept of program correctness, the differences between high-level languages, assembly languages and machine languages, the role played by compilers, the execution of programs by computer hardware. Together with ISYS114 Introduction to Systems Design and Data Management, this unit forms the entry point for mainstream computing units.

### **Assumed Knowledge**

COMP115 does not have any particular assumed knowledge.

## Expected Learning Outcomes

Students successfully completing COMP115 will be able to:

1. Describe the main components of a computer system and the role that different kinds of programming language play in computer software development.
2. Apply problem solving skills to develop algorithms that solve small to medium-sized computational problems.
3. Design and code implementations of their algorithms in an imperative programming language.
4. Use standard software engineering practices to document, debug and test their programs.
5. Identify and describe ethical issues that arise in the application of information technology.

All assessment tasks will involve problem solving and analysis, and the solutions require the design of simple algorithms. Programs will be written in the Processing language using a range of standard imperative programming language features. In developing and assessing practical work, particular attention is given to software engineering principles, most notably documentation (suitable names for variables and functions, suitable use of comments), testing and debugging.

## Graduate Capabilities Developed

The COMP115 learning outcomes develop general graduate capabilities, including

- *Discipline Specific Knowledge and Skills (outcomes 1-5)*
- *Critical, Analytical and Integrative Thinking (outcomes 2-5)*
- *Problem Solving and Research Capability (outcomes 2-4)*
- *Creativity and Innovation (outcomes 2-4)*
- *Effective Communication (outcomes 1, 4 and 5)*

COMP115 provides discipline specific knowledge about software execution, specification and development with practical skills in programming using the Processing language along with an awareness of ethical issues that arise in information technology. Problem solving, algorithm design and programming based on informal specifications require limited research of different methods, critical analysis of these methods and an ability to integrate methods into a solution and explain it to others.

## Classes and Workload

Each week of COMP115 has three hours of lectures and a two-hour practical class. For details of days, times and rooms, consult the University timetables webpage (<http://www.timetables.mq.edu.au>). The Day and Evening streams of COMP115 have the same content. Practical classes commence in Week 1 and are held in the E6A Computer Laboratories.

Audio and screen video recordings of the lectures in the Day stream will be made available online via iLecture (<http://online.mq.edu.au/docs/qgilec.html>).

Macquarie University generally expects students to need to work for about three hours per credit point each week. Since COMP115 is a 3 credit point unit, it should therefore require about nine hours per week for average students. There are five hours per week of classes during teaching weeks; therefore an average student should expect to spend about four hours out of class per week on COMP115 during teaching weeks. In the lecture break when there are no scheduled classes, so that time should be used for additional study.

## Online Resources

COMP115 Web Home Page: <http://www.comp.mq.edu.au/units/comp115/>

As part of a trial in the Department of Computing, COMP115 will make extensive use of the Moodle course management system, including for delivery of class materials, discussion boards, online self-tests, submission of work and access to marks and feedback. Students should check the Moodle site (<https://moodle.comp.mq.edu.au/>) regularly for unit updates. Instruction on the use of Moodle in COMP115 will be provided. Because COMP115 is using Moodle, the unit will not be using the University's Online Learning @ MQ site (WebCT/Blackboard).

Questions and general queries regarding the content of this unit, its lectures or practicals should be posted to the appropriate discussion board on the COMP115 Moodle site. 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. Questions of a private nature should be directed to the unit teaching staff.

## Technology Used and Required

The practical work in this unit involves programming in the Processing language (<http://processing.org>) which will give students experience with features that are used in many modern programming languages. The Processing software can be downloaded free of charge for Windows, Linux and Mac OS X computers from the Processing web site. It is also installed in the E6A Computer Laboratories.

## Required Unit Materials and Recommended Readings

*Learning Processing: A Beginner's Guide to Programming Images, Animation, and Interaction* by Daniel Shiffman, Morgan Kaufmann, 2008.

We cover a large proportion of the material in this book and it will be difficult to successfully complete this unit without reading the relevant chapters. You will find the lecture material much easier to understand if you read the textbook in advance of the lectures. The lecture schedule below lists the relevant sections of the textbook.

The textbook website at <http://www.learningprocessing.com/> provides supplementary material that you may find useful, including tutorials on Processing, the complete code for the examples in the book, and related downloads.

The Macquarie University library has a number of copies of the textbook, including some in the reserve collection. The library also has many other books on programming that you

may find useful if the concepts are not adequately explained by the textbook or class material.

## Teaching Strategy

COMP115 is taught via lectures and practical classes in the laboratory. Lectures are used to introduce new material, give examples of programming language constructs and their implementations, put this material in a wider context, and to go through exercises that require a detailed knowledge of the lecture material. You are strongly encouraged to ask questions of the lecturer to clarify anything you might not be sure about, particularly while exercises are being solved.

The practical sessions give you an opportunity to practice your programming skills under the supervision of a tutor. Each week you will be given problems to work on; it is important that you keep up with these problems as doing so will help you understand the material in the unit and prepare you for the assignment work. You will be asked to have your solution to one programming problem each week checked by your tutor in class, for which you will receive marks. Some practical classes will also feature written exercises to prepare you for the examination.

In summary, each week you should:

- Read lecture notes and book chapters prior to attending the lectures on that material.
- Attend lectures, take notes, ask questions.
- Submit answers to assigned questions.
- Attend your practical classes, seeking feedback from your tutor on your submitted work.
- Work on the practical exercises and assignments.

Lecture notes or slides will be made available but they are intended as an outline of the lecture only and are not a substitute for attending the lecture, making your own notes and reading the textbook.

## Lecture Plan

The following table lists the main topics to be covered in each week of COMP115 lectures. Relevant textbook chapters are listed. In addition, material from chapters 11 and 12 will be covered throughout the unit.

<i>Week</i>	<i>Topic</i>	<i>Lecturer</i>	<i>Textbook</i>
1	Introduction to Computer Science	Sloane	Introduction, Chapter 2
2	Pixels, Interaction and Variables	Sloane	Chapters 1, 3, 4
3	Conditionals and Loops	Sloane	Chapters 5, 6
4	Loops	Sloane	Chapter 6
5	Functions	Verity	Chapter 7
6	Design	Verity	Chapter 10

<i>Week</i>	<i>Topic</i>	<i>Lecturer</i>	<i>Textbook</i>
7	Objects	Verity	Chapter 8
	<i>Lecture Recess - Two Weeks</i>		
8+9	Arrays; Text	Sloane	Chapters 9, 17
10+11	Data Input, XML and the Web	Verity	Chapters 18
12	Problem solving; Ethics	Verity	
13	Review, Exam Discussion	Sloane Verity	

### Assessment Tasks

<b>Task</b>	<b>Duration</b>	<b>Learning Outcomes Assessed</b>	<b>Weight</b>
Weekly Exercises	Weeks 1-12	1 to 4	10%
Assignment One	Weeks 2-4	2 to 4	3%
Assignment Two	Weeks 5-8	2 to 4	15%
Assignment Three	Weeks 9-12	2 to 4	12%
Final Examination	3 hours	All	60%

In practical classes, you will be asked to have your tutor check your solution to a short programming exercise. Completing a solution to the exercise should take no more than 30-45 minutes if you are keeping up with the lecture material. Each exercise assessed in this way is worth 1% of the assessment with a maximum of 10% counting toward your final assessment.

The assignments are programming exercises that allow skills to be demonstrated by solving a more substantial problem than in the weekly exercises. The first assignment is a straight-forward exercise to make sure that you are familiar with the basics of Processing programming early in the unit. The other two assignments are a significant amount of work and will involve applying unfamiliar ideas to practical problems. Hence you are unlikely to be able to do them in the week before they are due. Make sure you leave enough time to understand the relevant ideas and code before you try to solve the assignment.

You are encouraged to:

- set your personal deadline earlier than the actual one;
- keep backups of all your important files;
- make sure that no-one else picks up your printouts;
- do not otherwise share your code with other students.

Late work will not be accepted. If you cannot submit on time because of illness or other circumstances, please contact the lecturer before the due date.

The weekly practical exercises will be submitted and assessed in your classes with immediate feedback from your tutor. Assignments will be submitted online via the Moodle site. Feedback about assignments should be provided within two weeks of submission, in the form of an assessed mark, an explanation of how that mark was determined from the submission and general advice about strengths and weaknesses in the work.

A closed-book three hour written final examination will be held in the examination period at the end of the semester (6-24 June). The aim of the examination is to determine the extent to which each student has achieved the learning outcomes of the unit.

To help students assess their progress toward the final examination and to provide feedback to unit staff on that progress, there will be optional online progress tests that do not count towards the final assessment. We will also conduct some written exercises in practical classes to give you practice at answering the type of question that will appear on the examination.

### **Assessment Standards**

COMP115 will be graded according to the following general descriptions of the letter grades as specified by Macquarie University.

- **High Distinction (HD, 85-100):** provides consistent evidence of deep and critical understanding in relation to the learning outcomes. There is substantial originality and insight in identifying, generating and communicating competing arguments, perspectives or problem solving approaches; critical evaluation of problems, their solutions and their implications; creativity in application as appropriate to the discipline.
- **Distinction (D, 75-84):** provides evidence of integration and evaluation of critical ideas, principles and theories, distinctive insight and ability in applying relevant skills and concepts in relation to learning outcomes. There is demonstration of frequent originality in defining and analysing issues or problems and providing solutions; and the use of means of communication appropriate to the discipline and the audience.
- **Credit (Cr, 65-74):** provides evidence of learning that goes beyond replication of content knowledge or skills relevant to the learning outcomes. There is demonstration of substantial understanding of fundamental concepts in the field of study and the ability to apply these concepts in a variety of contexts; convincing argumentation with appropriate coherent justification; communication of ideas fluently and clearly in terms of the conventions of the discipline..
- **Pass (P, 50-64):** provides sufficient evidence of the achievement of learning outcomes. There is demonstration of understanding and application of fundamental concepts of the field of study; routine argumentation with acceptable justification; communication of information and ideas adequately in terms of the conventions of the discipline. The learning attainment is considered satisfactory or adequate or competent or capable in relation to the specified outcomes.
- **Fail (F, 0-49):** does not provide evidence of attainment of learning outcomes. There is missing or partial or superficial or faulty understanding and application of the fundamental concepts in the field of study; missing, undeveloped, inappropriate or

confusing argumentation; incomplete, confusing or lacking communication of ideas in ways that give little attention to the conventions of the discipline.

The standards of achievement that will be used to assess each of the assessment tasks with respect to the letter grades are as follows.

*Learning Outcome 1: Describe the main components of a computer system and the role that different kinds of programming language play in computer software development.*

*Learning Outcome 5: Identify and describe ethical issues that arise in the application of information technology.*

<b>P</b>	Can correctly reproduce basic facts and definitions across a breadth of concepts and issues, but lacks depth of understanding.
<b>Cr or D</b>	Exhibits breadth and depth of understanding of concepts and issues. Can use terminology accurately in new contexts. Can express ideas in their own words and has an understanding of the limits of their understanding.
<b>HD</b>	As for Cr or D and is aware of the context in which the concepts and issues are developed and their limitations. Able to generate and justify principles and hypotheses for existing or new concepts or issues.

*Learning Outcome 2: Apply problem solving skills to develop algorithms that solve small to medium-sized computational problems.*

<b>P</b>	Can develop algorithms for problems that are similar to provided examples.
<b>Cr or D</b>	Can analyse problems that differ from provided examples and apply a variety of provided algorithmic approaches to their solution.
<b>HD</b>	As for Cr or D, and can recognise the limitations of known algorithmic approaches and is able to develop alternatives.

*Learning Outcome 3: Design and code implementations of their algorithms in an imperative programming language.*

<b>P</b>	Can implement basic algorithms based on similar provided examples.
<b>Cr or D</b>	As for P and can use a wide range of provided programming language features to implement algorithms whose detailed implementation has not previously been discussed.
<b>HD</b>	As for Cr or D and can develop programs using techniques or approaches that have not been discussed.

*Learning Outcome 4: Use standard software engineering practices to document, debug and test their programs.*

<b>P</b>	Can apply some basic documentation, debugging and testing practices along the lines of examples provided.
<b>Cr or D</b>	Is able to apply a wide range of documentation, debugging and testing practices to their code along the lines of examples provided.
<b>HD</b>	As for Cr or D and has well-developed skills for applying documentation, debugging and testing practices in ways that have not been previously illustrated by examples.

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 raw 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.

On occasion your raw mark for the unit may not be the same as the Standardised Numeric Grade (SNG) which you receive as the final result. Under the Senate guidelines, raw marks may be scaled to ensure that there is a degree of comparability across the university, so that units with the same past performances of their students should achieve similar results.

### **Extension Requests and Special Consideration**

The only exception to not completing an assessment task in the time allocated or not sitting an examination at the designated time is because of serious documented illness or unavoidable disruption. In these circumstances you may wish to consider applying for Special Consideration. Information about unavoidable disruption and the special consideration process is available in the University Special Consideration policy ([http://www.mq.edu.au/policy/docs/special\\_consideration/policy.html](http://www.mq.edu.au/policy/docs/special_consideration/policy.html)). The relevant form is on the web (<http://www.registrar.mq.edu.au/Forms/APScons.pdf>).

If you apply for Special Consideration and it is judged by the Department of Computing that your performance on an examination has been affected adversely by the circumstances documented in the consideration request, you will be required to sit a Supplementary Examination. The Supplementary Examination will normally be scheduled after the conclusion of the official examination period, but may be earlier in the case of a mid-semester examination. For details of the Special Consideration policy specific to the Department of Computing, see the Department's special consideration policy page ([http://www.comp.mq.edu.au/undergrad/policies/special\\_consideration\\_policy.htm](http://www.comp.mq.edu.au/undergrad/policies/special_consideration_policy.htm)).

### **Changes to COMP115 since the Last Offering**

This offering of COMP115 is the second one to use the Processing programming language. No major changes are being made to this version, except that the mixed tutorial and practical classes previously used are being replaced with practical classes so that students get more chance to practice their programming with tutor help. Tutorial exercises will be incorporated into lecture times.

## Useful Resources

All students should familiarize themselves with the policies that govern learning and teaching at Macquarie University, in particular:

- Macquarie University Academic Honesty Policy  
[http://www.mq.edu.au/policy/docs/academic\\_honesty/policy.html](http://www.mq.edu.au/policy/docs/academic_honesty/policy.html)
- Macquarie University Assessment Policy and Code of Practice  
<http://www.mq.edu.au/policy/docs/assessment/policy.html>  
[http://www.mq.edu.au/policy/docs/assessment/policy\\_code\\_of\\_practice.html](http://www.mq.edu.au/policy/docs/assessment/policy_code_of_practice.html)
- Department of Computing Special Consideration Policy  
[http://www.comp.mq.edu.au/undergrad/policies/special\\_consideration\\_policy.htm](http://www.comp.mq.edu.au/undergrad/policies/special_consideration_policy.htm)
- Macquarie University Special Consideration Policy  
[http://www.mq.edu.au/policy/docs/special\\_consideration/policy.html](http://www.mq.edu.au/policy/docs/special_consideration/policy.html)

HELP101 is a service provided by the Department of Computing where tutors are available outside class time to answer questions. If you have a general question, any of the tutors may be able to help you; if your question is specific to COMP115, you will need to find the COMP115 tutor. Consult the HELP101 timetable in the laboratories for details of tutor availability.

Enquires relating specifically to COMP115 that can't be addressed by a tutor should be directed to the Unit Convenor or Lecturer.

The Department of Computing has established a Staff-Student Liaison Committee at each level (100, 200, 300) to provide all students studying a Computing unit the opportunity to discuss related issues or problems with both fellow students and staff. Information about the 100-level Liaison Committee will be made available in lectures and on the Web at:

<http://www.comp.mq.edu.au/undergrad/info/liaison/100-level/>

Macquarie University also provides a range of Academic Student Support Services. Details of these services can be accessed at <http://www.student.mq.edu.au>.

If all other avenues have been exhausted, serious issues relating to study in the Department of Computing in general or in this unit in particular should be raised with the Department's Director of Teaching (Dr. Christophe Doche) or the Head of Department (Prof. Bernard Mans).