COMP115

Introduction to Computer Programming

S1 Evening 2018

Dept of Computing

Contents

General Information 2

Learning Outcomes 3

General Assessment Information 3

Assessment Tasks 3

Delivery and Resources 6

Unit Schedule 7

Policies and Procedures 8

Graduate Capabilities 10

Changes from Previous Offering 13

Assessment Standards 14

Changes since First Published 15

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
Lecturer, Convener
Matthew Roberts
matthew.roberts@mq.edu.au
Contact via matthew.roberts@mq.edu.au
E6A374
Tue 12-2pm

Lecturer
Young Lee
young.lee@mq.edu.au
Contact via young.lee@mq.edu.au
E6A376
Mon 2-3pm (weeks 7-12)

Lecturer
Dominic Verity
dominic.verity@mq.edu.au
Contact via dominic.verity@mq.edu.au
E6A360
Mon 1-2pm, Fri 2-3pm (weeks 1-4)

Credit points
3

Prerequisites

Corequisites

Co-badged status

Unit description
This unit is an introductory computer science unit, providing a practical introduction to basic computing and programming concepts. Students 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; and 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.

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:

Describe the main components of a computer system and the role that different kinds of programming language play in computer software development

Apply problem solving skills to develop algorithms that solve small to medium-sized computational problems

Design and code implementations of their algorithms in an imperative programming language

Use standard software engineering practices to document, debug and test their programs

Identify and describe ethical issues that arise in the application of information technology

Demonstrate foundational learning skills particular to computer programming including active engagement in their learning process.

General Assessment Information

Assessment Tasks

Name	Weighting	Hurdle	Due
Assignment One	12%	No	Weeks 3-7

Name	Weighting	Hurdle	Due
In-class Exercises	12%	Yes	Weeks 2-12
Assignment Two	16%	No	Weeks 7-11
Module Exams	60%	Yes	Various

Assignment One

Due: **Weeks 3-7**
Weighting: **12%**

The assignments are programming exercises that allow skills to be demonstrated by solving a more substantial problem than in the weekly exercises. Assignment One is a relatively simple exercise that is designed to begin building competency in using the Processing language to solve problems.

On successful completion you will be able to:

- Apply problem solving skills to develop algorithms that solve small to medium-sized computational problems
- Design and code implementations of their algorithms in an imperative programming language
- Use standard software engineering practices to document, debug and test their programs

In-class Exercises

Due: **Weeks 2-12**
Weighting: **12%**
This is a hurdle assessment task (see assessment policy for more information on hurdle assessment tasks)

In each workshop class, besides the first and those that have exams, you will be graded on your preparation for and participation in the class. Active participation in the class will attract 4 marks of 10. 10 marks out of 10 are reserved for students who can demonstrate active participation and mastery of that week's material. To prepare for class and demonstrate mastery you should read the course readings and prepare answers to the homework exercises.

NB: This assessment represents the fact that laboratory classes are considered compulsory in this course.

Students must demonstrate satisfactory performance on the in-class exercises to pass the course. Satisfactory performance is defined as getting more than 40% on at least five of the of the eight in-class exercises.

On successful completion you will be able to:

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

Assignment Two

Due: **Weeks 7-11**
Weighting: **16%**

Assignment Two builds on the first assignment and the practical work to reinforce the basics of programming and includes some of the more advanced aspects that are covered in the second half of the unit.

On successful completion you will be able to:

- Apply problem solving skills to develop algorithms that solve small to medium-sized computational problems
- Design and code implementations of their algorithms in an imperative programming language
- Use standard software engineering practices to document, debug and test their programs

Module Exams

Due: **Various**
Weighting: **60%**
This is a hurdle assessment task (see assessment policy for more information on hurdle assessment tasks)

The module examinations ask students to answer conceptual questions about the course material as well as solve simple programming problems. Module exams are run in the first hour of the workhop in which the student is enrolled. Students may only attend module exams in workshops they are enrolled in. In the case a student cannot attend a module exam, a request for special consideration must be made.

Six module examinations are each offerred up to three times during semester. The student's best mark for each module is used in their final mark.

The exam mark for each module is worth 10% of the final mark.

Students must demonstrate satisfactory performance in modules 2, 3, 4 and 5 to pass the course. Satisfactory performance is defined as getting more than 40% on at least one of the module exams offered for that module. A student's final mark for a module is the maximum mark they achieved in any of the module exams for that module.

On successful completion you will be able to:

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

Delivery and Resources

CLASSES

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.

REQUIRED AND RECOMMENDED TEXTS AND/OR MATERIALS

- *Learning Processing: A Beginner's Guide to Programming Images, Animation, and Interaction*, Daniel Shiffman, Morgan Kaufmann, 2nd edition, 2015

We cover a large proportion of the material in this book and it will be extremely difficult to successfully complete this unit without reading the relevant chapters. This text is the primary source of examinable material in this unit. Furthermore, 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.

UNIT WEBPAGE AND TECHNOLOGY USED AND REQUIRED

Web Home Page

COMP115 will make extensive use of the iLearn 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 iLearn site (https://ilearn.mq.edu.au) regularly for unit updates.

Questions and general queries regarding the content of this unit, its lectures or practical classes, or its assessments should be posted to the discussion boards on the COMP115 iLearn 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.

iLecture

Any audio and screen video recordings of the lectures in the Day and Evening streams will be made available online at iLearn via the echo360 system.

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.

There are many versions of Processing available. In our teaching and in the laboratories we will be using version 3.0.1. To ensure compatibility with us you should make sure to install this version, not an older version, nor a newer version if one should become available during the semester.

Unit Schedule

The unit is broken into six modules

1. Foundations of Programming (FP)
2. Variables and Conditionals (V&C)
3. Loops (L)
4. Functions (F)
5. Arrays and Strings (A&S)

6. Program Design and Problem Solving (PD&PS)

All but the first and last modules (V&C, L, F, A&S) cover one skill which is absolutley necessary to program a computer. Thus all students must demonstrate a satisfactory performance in each of these modules to pass the course. Satisfactory performance in a module is defined in the assessment tasks section. The first and final modules (FP, PD&PS) introduce then synthesise the skills learned in the other modules. Student performance in these module is important to their final grade and to demonstrate they have reached the level of mastery required to pass, but less than satisfactory performance in this module does not preclude a student from passing (as it does for the other modules).

Module exams are offerred five times during semester giving students more than one opportunity to demonstrate satisfactory performance in each module.

Week	Topic	Module Exams	Lecturer	Textbook
1	Foundations of Programming 1		Dom	Chapters 1,2, Lecture Notes
2	Foundations of Programming 2		Dom	Chapters 1,2, Lecture Notes
3	Variables and Conditions 1		Dom	Chapters 3,4,5
4	Variables and Conditions 2		Matt	Chapters 3,4,5
5	Loops 1		Matt	Chapter 6
6	Loops 2		Matt	Chapter 6
7	Program Design and Problem Solving 1	FP, V&C	Matt/Young	Chapter 10, Lecture Notes
	Mid Semester Break			
8	Functions 1	V&C, L	Matt/Young	Chapter 7
9	Functions 2		Matt/Young	Chapter 7
10	Arrays and Strings 1		Matt/Young	Chapters 9, 17
11	Arrays and Strings 2		Matt/Young	Chapters 9, 17
12	Program Design and Problem Solving 2	L, F, A&S	Matt/Young	Lecture notes
13	Review; Exam Preparation	F, A&S, PD&PS	TBA	
Exam Period		FP, V&C, L, F, A&S, PD&PS		

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
- 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.)*

Undergraduate 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 shown in *iLearn*, 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.

If you receive special consideration for the final exam, a supplementary exam will be scheduled in the interval between the regular exam period and the start of the next session. By making a special consideration application for the final exam you are declaring yourself available for a resit during the supplementary examination period and will not be eligible for a second special consideration approval based on pre-existing commitments. Please ensure you are familiar with the policy prior to submitting an application. You can check the supplementary exam information page on FSE101 in iLearn (bit.ly/FSESupp) for dates, and approved applicants will receive an individual notification one week prior to the exam with the exact date and time of their supplementary examination.

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 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

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.

Graduate Capabilities

Creative and Innovative

Our graduates will also be capable of creative thinking and of creating knowledge. They will be imaginative and open to experience and capable of innovation at work and in the community. We want them to be engaged in applying their critical, creative thinking.

This graduate capability is supported by:

Learning outcomes

- Apply problem solving skills to develop algorithms that solve small to medium-sized computational problems
- Design and code implementations of their algorithms in an imperative programming language
- Use standard software engineering practices to document, debug and test their programs

Assessment tasks

- Assignment One
- In-class Exercises
- Assignment Two

- Module Exams

Commitment to Continuous Learning

Our graduates will have enquiring minds and a literate curiosity which will lead them to pursue knowledge for its own sake. They will continue to pursue learning in their careers and as they participate in the world. They will be capable of reflecting on their experiences and relationships with others and the environment, learning from them, and growing - personally, professionally and socially.

This graduate capability is supported by:

Learning outcome

- Demonstrate foundational learning skills particular to computer programming including active engagement in their learning process.

Assessment task

- In-class Exercises

Discipline Specific Knowledge and Skills

Our graduates will take with them the intellectual development, depth and breadth of knowledge, scholarly understanding, and specific subject content in their chosen fields to make them competent and confident in their subject or profession. They will be able to demonstrate, where relevant, professional technical competence and meet professional standards. They will be able to articulate the structure of knowledge of their discipline, be able to adapt discipline-specific knowledge to novel situations, and be able to contribute from their discipline to inter-disciplinary solutions to problems.

This graduate capability is supported by:

Learning outcomes

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

Assessment tasks

- Assignment One

- In-class Exercises
- Assignment Two
- Module Exams

Critical, Analytical and Integrative Thinking

We want our graduates to be capable of reasoning, questioning and analysing, and to integrate and synthesise learning and knowledge from a range of sources and environments; to be able to critique constraints, assumptions and limitations; to be able to think independently and systemically in relation to scholarly activity, in the workplace, and in the world. We want them to have a level of scientific and information technology literacy.

This graduate capability is supported by:

Learning outcomes

- Apply problem solving skills to develop algorithms that solve small to medium-sized computational problems
- Design and code implementations of their algorithms in an imperative programming language
- Use standard software engineering practices to document, debug and test their programs
- Identify and describe ethical issues that arise in the application of information technology
- Demonstrate foundational learning skills particular to computer programming including active engagement in their learning process.

Assessment tasks

- Assignment One
- In-class Exercises
- Assignment Two
- Module Exams

Problem Solving and Research Capability

Our graduates should be capable of researching; of analysing, and interpreting and assessing data and information in various forms; of drawing connections across fields of knowledge; and they should be able to relate their knowledge to complex situations at work or in the world, in order to diagnose and solve problems. We want them to have the confidence to take the initiative in doing so, within an awareness of their own limitations.

This graduate capability is supported by:

Learning outcomes

- Apply problem solving skills to develop algorithms that solve small to medium-sized

computational problems
- Design and code implementations of their algorithms in an imperative programming language
- Use standard software engineering practices to document, debug and test their programs
- Demonstrate foundational learning skills particular to computer programming including active engagement in their learning process.

Assessment tasks

- Assignment One
- In-class Exercises
- Assignment Two
- Module Exams

Effective Communication

We want to develop in our students the ability to communicate and convey their views in forms effective with different audiences. We want our graduates to take with them the capability to read, listen, question, gather and evaluate information resources in a variety of formats, assess, write clearly, speak effectively, and to use visual communication and communication technologies as appropriate.

This graduate capability is supported by:

Learning outcomes

- Describe the main components of a computer system and the role that different kinds of programming language play in computer software development
- Use standard software engineering practices to document, debug and test their programs
- Identify and describe ethical issues that arise in the application of information technology

Assessment tasks

- Assignment One
- In-class Exercises
- Assignment Two
- Module Exams

Changes from Previous Offering

New module structure, one fewer compulsory modules and the addition of compulsory practical work.

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.

P	Can correctly reproduce basic facts and definitions across a breadth of concepts and issues, but lacks depth of understanding.
Cr or D	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.

HD	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.

P	Can develop algorithms for problems that are similar to provided examples.
Cr or D	Can analyse problems that differ from provided examples and apply a variety of provided algorithmic approaches to their solution.
HD	As for Cr or D and can develop programs using techniques or approaches that have not been discussed.

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

P	Can implement basic algorithms based on similar provided examples.
Cr or D	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.
HD	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.

P	Can apply some basic documentation, debugging and testing practices along the lines of examples provided.
Cr or D	Is able to apply a wide range of documentation, debugging and testing practices to their code along the lines of examples provided.
HD	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.

We will look at your overall performance on all assessments when determining your final grade. A total raw mark of at least 50% and satisfactory performance in modules 2, 3, 4, 5 and the in-class exercises will be sufficient to pass the unit.

Changes since First Published

Date	Description
21/02/2018	updated schedule to reflect exam changes due to easter holidy