

INFT1004: Introduction to Programming

Callaghan and Ourimbah

Semester 1 - 2017



OVERVIEW

Course Description

Introduces the key skills of problem solving and computer programming, including the elementary programming concepts of documentation, data elements, sequence, selection, and iteration.

Assumed Knowledge Contact Hours

None.

Computer Lab

Face to Face On Campus
2 hour(s) per Week for Full Term

Lecture

Face to Face On Campus
2 hour(s) per Week for Full Term

Unit Weighting Workload

10

Students are required to spend on average 120-140 hours of effort (contact and non-contact) including assessments per 10 unit course.

COURSE OUTLINE

CONTACTS

Course Coordinator

Callaghan

Dr Keith Nesbitt

Keith.Nesbitt@newcastle.edu.au

(02) 4985 4519

Consultation: Tuesday – 2:00pm - 3pm (before lecture – or by appointment)

Teaching Staff

Other teaching staff will be advised on the course Blackboard site.

School Office

School of Electrical Engineering and Computing

EAG08

EA Building

Callaghan

+61 2 4921 6026

8.30am- 12.30pm and 1.30pm-4.30pm (Mon-Fri)

SYLLABUS

Course Content

1. Programming language syntax
2. Elementary programming concepts and structures
3. Problem analysis techniques
4. Program design and development
5. Documentation techniques
6. Testing and debugging techniques
7. Use of programs to manipulate multimedia files

Course Learning Outcomes

On successful completion of this course, students will be able to:

1. Comprehend the concepts of Object-Oriented and event-driven programming;
2. Comprehend a programming problem and design a solution;
3. Code a solution to a problem;
4. Test and document program solutions.

Course Materials

Lecture materials:

- Provided online via Blackboard

Suggested text:

- Mark J Guzdial & Barbara Ericson, Introduction to Computing and Programming in Python; a Multimedia Approach, fourth edition. Prentice Hall, 2016.

Other resources:

- JES. This software is installed in the computer labs. For installation on students' own computers, JES 5.02 can be downloaded free from the Media Computation website, <http://mediacomputation.org>.

SCHEDULE - Callaghan

Week	Week Begins	Topic	Learning Activity	Assessment Due
1	27 Feb	Introduction, Assignment, Arithmetic	Module 1.1, 1.2, 1.3, 1.4	
2	6 Mar	Sequence, Quick Start, Program Style	Module 2.1, 2.2, 2.3	
3	13 Mar	Pictures, Functions, Media Paths	Module 3.1, 3.2, 3.3	
4	20 Mar	Arrays, Pixels, For Loop, Ref Passing	Module 4.1, 4.2, 4.3	
5	27 Mar	Nested Loops, Selection, More Pictures	Module 5.1, 5.2, 5.3	Practical Exam
6	3 Apr	Lists, Strings, Input & Output, Files	Module 6.1, 6.2, 6.3, 6.4	
7	10 Apr	Drawing Pictures, Program Design, While	Module 7.1, 7.2, 7.3	Assign Part 1, 2 set
Mid Semester Break				
8	24 Apr	No Lecture / Revision in Lab		
9	1 May	Data Structures, Processing sound	Module 9.1, 9.2	
10	8 May	Advanced Sound	Module 10.1	Assign Part 1 Due
11	15 May	Making Movies, Scope, Import	Module 11.1, 11.2	
12	22 May	Turtles, Writing Classes	Module 12.1, 12.2	Assign Part 2 Due
13	29 May	Revision	Module 13.1	
Semester 1 Examinations Week 1				
Semester 1 Examinations Week 2				
Semester 1 Examinations Week 3				

2017 - Retention of Assignment Scripts

In 2018, the University of Newcastle will undergo its 5 yearly accreditation cycle with Engineers Australia. Part of this routine process is the collection of a sample of student assignments over a 1+ year period. The objective is to provide the accrediting panel an indication of educational rigour across all courses. In 2017, we will commence collecting sample assignments from each and every assessment task, across the full spectrum of marks. They will be provided to the panel but not de-identified. If you object to your assignment being retained with your name associated, please indicate this on the submission, and if retained, we will de-identify your paper. All papers will be destroyed at the completion of the accreditation process.

ASSESSMENTS – Callaghan

This course has 4 assessments. Each assessment is described in more detail in the sections below.

	Assessment Name	Due Date	Involvement	Weighting	Learning Outcomes
1	Quiz	most weeks, in lab class	Individual	Formative	1, 2, 3
2	Programming tests	week 5, in lab class	Individual	Formative	1, 2, 3, 4
3	Programming assignment	part 1: week 10 part 2: week 12	Pair	20% 30%	1, 2, 3, 4
4	Final Examination	formal examination period	Individual	50%	1, 2, 3

Late Submissions

The mark for an assessment item submitted after the designated time on the due date, without an approved extension of time, will be reduced by 10% of the possible maximum mark for that assessment item for each day or part day that the assessment item is late. Note: this applies equally to week and weekend days.

Assessment 1 – Quizzes

Assessment Type	Examination – class
Description	Online quizzes. These are formative assessment; they help you to establish how well you are progressing, but do not contribute to your final mark.
Weighting	zero
Due Date	Most weeks, in lab class
Submission Method	Blackboard
Assessment Criteria	The quiz will be multiple-choice, and will be marked according to the number of questions answered correctly.
Return Method	Not returned
Feedback Provided	Marks will be available shortly after the quiz; individual feedback will not be provided.

Assessment 2 – Practical programming test

Assessment Type	Examination – class
Description	The practical programming test consists of a discrete programming task which students are required to complete in tutorial / lab class. The test will provide the students with early feedback on their learning, highlighting any areas of concern and possibly stimulating discussion with tutors and lecturers. This is formative assessment, and does not contribute to your final mark for the course.
Weighting	zero
Due Date	Week 5, in lab class
Submission Method	Blackboard
Assessment Criteria	Correctness, functionality, and style of code
Return Method	Self-assessment.
Feedback Provided	General Feedback once the tests have all been completed. Individual feedback will not be provided.

Assessment 3 – Programming assignment

Assessment Type	Written assignment
Description	The programming assignment permits students to develop a comprehensive program over the space of several weeks, to thoroughly test and document it, with the documentation including a contemporaneous journal. It also gives students the opportunity to develop software in a small group (specifically, a pair).
Weighting	50%; 20% for part 1, 30% for part 2
Due Date	Week 10 (part 1); week 12 (part 2)
Submission Method	Blackboard; a signed cover sheet is also required
Assessment Criteria	Correctness, functionality, and style of code, and completeness of accompanying journal
Return Method	Feedback once the assignments have all been marked
Feedback Provided	Yes

Assessment 4 – Final Examination

Assessment Type	Formal examination
Description	The final formal examination is designed to test the individual students' knowledge of the course material and their ability to describe, analyse and hypothesise from this material.
Weighting	50%
Due Date	As timetabled in formal examination period
Submission Method	As a formal written exam
Assessment Criteria	Correctness of answers
Return Method	Not returned
Feedback Provided	No

ADDITIONAL INFORMATION

Grading Scheme

This course is graded as follows:

Range of Marks	Grade	Description
85-100	High Distinction (HD)	Outstanding standard indicating comprehensive knowledge and understanding of the relevant materials; demonstration of an outstanding level of academic achievement; mastery of skills*; and achievement of all assessment objectives.
75-84	Distinction (D)	Excellent standard indicating a very high level of knowledge and understanding of the relevant materials; demonstration of a very high level of academic ability; sound development of skills*; and achievement of all assessment objectives.
65-74	Credit (C)	Good standard indicating a high level of knowledge and understanding of the relevant materials; demonstration of a high level of academic achievement; reasonable development of skills*; and achievement of all learning outcomes.
50-64	Pass (P)	Satisfactory standard indicating an adequate knowledge and understanding of the relevant materials; demonstration of an adequate level of academic achievement; satisfactory development of skills*; and achievement of all learning outcomes.
0-49	Fail (FF)	Failure to satisfactorily achieve learning outcomes. If all compulsory course components are not completed the mark will be zero. A fail grade may also be awarded following disciplinary action.

Attendance

*Skills are those identified for the purposes of assessment task(s).

Attendance/participation will be recorded in the following components:

- Computer Lab (Method of recording:)
- Lecture (Method of recording:)

Communication Methods

Communication methods used in this course include:

- Blackboard Course Site: Students will receive communications via the posting of content or announcements on the Blackboard course site.
- Email: Students will receive communications via their student email account.
- Face to Face: Communication will be provided via face to face meetings or supervision.

Course Evaluation

Each year feedback is sought from students and other stakeholders about the courses offered in the University for the purposes of identifying areas of excellence and potential improvement.

Most feedback on previous offerings of this course is positive. The negative comments tend to focus on the challenge of the course . . .

“Sometimes it seems like there is too much content too fast; if you stop you drown.”

“Extremely challenging but also very satisfying.”

“The course is challenging, however with enough study it can be completed.”

“For a first-year course this was intensely difficult. It felt like the teachers expected us to know how to program right from the start.”

This challenge is probably true of any first programming course, particularly an introductory one. The best way to deal with it is to program”. That is, writing code from the beginning. Any delay in understanding and applying the ideas can create a substantial roadblock to further progress.

On the other hand, some students believe that Python is not an appropriate language to be teaching:

“I don’t like the fact we learnt Python. It’s the least needed language in the workforce for programmers in 2013 and has been for years. Should switch to JavaScript or Objective C.”

This comment is largely correct. Although the purpose of this course is not necessarily to produce work-ready programmers. A principle goal is to teach the elements of programming to students who are not yet familiar with those elements. JES does provide a simple to install and use. It suffers slightly because it is not a strongly typed language and automatic reference passing. It is recognised that a more widely used language such as C# could have benefits, although it is also technically more difficult to start using. The choice of language for this course is constantly under review.

Academic Misconduct

All students are required to meet the academic integrity standards of the University. These standards reinforce the importance of integrity and honesty in an academic environment. Academic Integrity policies apply to all students of the University in all modes of study and in all locations. For the Student Academic Integrity policy, refer to <http://www.newcastle.edu.au/policy/000608.html>.

Adverse Circumstances

You are entitled to apply for special consideration because adverse circumstances have had an impact on your performance in an assessment item. This includes applying for an extension of time to complete an assessment item. Prior to applying you must refer to the Adverse Circumstances Affecting Assessment Items Procedure, available at <http://www.newcastle.edu.au/policy/000940.html>. All applications for Adverse Circumstances must be lodged via the online Adverse Circumstances system, along with supporting documentation.

Important Policy Information

The 'HELP for Students' tab in UoNline contains important information that all students should be familiar with, including various systems, policies and procedures.

R.H. Middleton

This course outline was approved by the Head of School. No alteration of this course outline is permitted without Head of School approval. If a change is approved, students will be notified and an amended course outline will be provided in the same manner as the original.

© 2017 The University of Newcastle, Australia