Course Code:	1001ICT
Course Name:	Programming 1
Semester:	Semester 1, 2017
Program:	Diploma of Engineering
	Diploma of Information Technology
Credit Points:	10
Course Coordinator:	Seyedali Mirjalili
Document modified:	02 DEC 2016

Teaching Team

Your lecturer/tutor can be contacted via the email system on the portal.

Staff Consultation

Your lecturer/tutor is available each week for consultation outside of normal class times. Times that your lecturer/tutor will be available for consultation will be given in the first week of lectures. A list of times and rooms will be published on the Griffith College Portal under the “myTimetable” link.

Prerequisites

Please note: 1001ICT is a prerequisite for courses 2002ICT, 2508ICT, 3642ICT and 1802ICT. This means that you need to achieve a Pass or above to progress to any of these courses. If you achieve only a PC, you will need to repeat 1001ICT before progressing to 2002ICT, 2508ICT, 3642ICT or 1802ICT.

Brief Course Description

Programming I is a 10 credit point course within the Diploma of Information Technology. The course is situated within the first semester of the program. The Diploma of Information Technology is designed to provide students with a pathway to:

- further university study in Information Technology and related degrees, or
- employment opportunities within the IT industry.

Programming I is the first programming course students encounter in the Diploma of Information Technolgy. The course introduces modern programming concepts and techniques and provides a foundation for subsequent programming courses within the Diploma of Information Technology.

Rationale

All information technology professionals require experience and skills in problem solving, and implementation of solutions on a computer. This course begins the development of these essential skills utilising mobile robotics. The approach taken is gently graduated, applied, and hands-on. The course introduces modern programming concepts and techniques that provide a foundation for futher refinement in subsequent programming courses.

Java is one of the most popular programming languages used by IT professionals today and it is a language within the object-oriented programming paradigm making it possible to easily progress to other widely-used languages. It is a full industry-strength language, suitable for large software engineering projects, but as such, it is helpful to introduce some of its features in simple stages first. For this we will use MaSH (Making Stuff Happen, a subset of the Java programming language with its own compiler developed specifically for this course). Any programming course will use problems drawn from various domains. The robotics area is growing in importance in industry, science and entertainment, and is more accessible and fun than most. Problems will also be drawn from non-robotics problem domains such as computer graphics.

Aims

Programming I aims to develop your understanding of basic programming concepts. As well, the course aims to develop your critical evaluation, analysis and problem solving skills through the exploration of problems and the implementation of practical programming solutions situated within the context of contemporary technology applications. The course also aims to develop your written, oral and interpersonal communication skills.

Learning Outcomes

Upon successful completion of this course you will be able to demonstrate:

1. an understanding of programming tools (editors and compilers);
2. an understanding of the properties of binary data, text data and the files that contain them;
3. how to interpret the formal specification of the syntax of a programming language;
4. how to create programs that consist of calls to procedures that perform actions;
5. how to interpret an API (Application Programming Interface);
6. how to create programs that declare variables, assign values to them, and call functions to return values;
7. how to create programs that involve repetition with definite and/or indefinite loops;
8. how to write programs that use selections;
9. how to write programs with multiple methods that call each other, sharing data with global variables and/or parameter passing;
10. how to create programs that use arrays;
11. how to document programs with comments; and
12. how to debug programs.
13. apply all of the above to solve problems with computer software.

Texts and Supporting Materials

There is no prescribed textbook. A recommended reading list will be provided via the course website. Where proprietary software is used for program development, it will be provided in laboratories on campus. Where possible, free and/or open source software that runs on Windows, Macintosh, and Linux will be used.

Organisation and Teaching Strategies

The course will be presented by a 2-hour lecture and a 2-hour laboratory per week. Lectures will be used to present problem solving and programming techniques which will be applied in subsequent laboratories, quizzes and the assignment. This course introduces you to modern programming techniques, including event handling and multi-threading.

The material is presented in a gently graduated manner, and uses an intuitive graphical programming environment to introduce concepts, before addressing them using more powerful traditional methods.

Class Contact Summary

Attendance:
Quizzes, assessed laboratories and the demonstration component of the project can only be done within formal class time, you are required to attend all lectures and their nominated laboratory session every week.

Your attendance in class will be marked twice during a four hour class. To receive full attendance, you must be present in the classroom on both occasions. Therefore, you are encouraged to attend and participate in all classes throughout the semester.

Participation in Class
You are expected to actively participate in classes each week.

Consultant Times
Attendance during consultation times is optional but you are encouraged to use this extra help to improve your learning outcomes.

Course Materials
Lecture notes will be made available to you on the Learning@Griffith College site on the student portal and you are expected to print these out and bring them to each class so that extra notes can be added.

Independent Learning
You are expected to reinforce your learning from class time by undertaking sufficient independent study {approximately 6 hours per week outside of class time} so that you can achieve the learning outcomes of the course. This may involve you spending time reviewing lecture notes, practicing programming skills and reviewing tutorial exercies.

Program Progression:
You are reminded that satisfactory Program Progression requires that attendance in classes is maintained at equal to or greater than 80%, and that GPA is maintained at equal to or greater than 3.5 [please see Griffith College Policy Library - Program Progression Policy - for more information].

Content Schedule

This course is comprised of five (5) interrelated themes:
1. programming concepts;
2. problem solving and design;
3. implementation in Lego Mindstorms NXT Software;
4. implementation in Java; and
5. use of programming tools and environments.

The main concepts emphasised include:
1. basic computer architecture;
2. virtual machines;
3. compilers and interpreters;
4. flow of control;
5. kinds of programming languages;
6. design approaches;
7. the programming process;
8. data storage; and
9. expressions.

Java is one of the most popular programming languages used by IT professionals today and it is a language within the object-oriented programming paradigm making it possible to easily progress to other widely-used languages. It is a full industry-strength language, suitable for large software engineering projects, but as such, it is helpful to introduce some of its features in a simpler language first. For this we will use Lego Mindstorms NXT Software. Any programming course will use problems drawn from various domains. The robotics area is growing in importance in industry, science and entertainment, and is more accessible and fun than most.

Weekly Teaching Schedule

Week	Topic	Activity	Readings
1	Introduction; Lego Mindstorms; Lego Mindstorms NXT Software; About Programming Languages; Elements of Java	Lecture	Refer to course web site
	Introduction & Demonstrations	Workshop	Refer to course web site
2	Lego Mindstorms NXT Software; Preparation for Quiz 1; A First MaSH Program; Syntax, Semantics and Compiler Phases	Lecture	Refer to course web site
	Lego Mindstorms NXT Software; A First MaSH Program	Tutorial	Refer to course web site
	Lab 1	Laboratory	
	Preparation for Quiz 1;	Workshop	
3	Console procedures; NXT procedures; Numbers	Lecture	Refer to course web site
	Console procedures; NXT procedures; Numbers	Tutorial	Refer to course web site
	Console procedures; NXT procedures; Numbers	Workshop	
	Lab 2	Laboratory	
4	Making stuff happen with numbers	Lecture	Refer to course web site
	Making stuff happen with numbers	Tutorial	Refer to course web site
	Making stuff happen with numbers	Workshop	
	Lab 3	Laboratory	
5	Debugging; Booleans; Java Booleans	Lecture	Refer to course web site
	Debugging; Booleans; Java Booleans	Tutorial	Refer to course web site
	Debugging; Booleans; Java Booleans	Workshop	
	Lab 4	Laboratory	

6	Math functions; Characters & Strings	Lecture	Refer to course web site
	Math functions; Characters & Strings	Tutorial	Refer to course web site
	Math functions; Characters & Strings	Workshop	
	Lab 5	Laboratory	
7	Compatibility and Conversions; Packaging into methods	Lecture	Refer to course web site
	Compatibility and Conversions; Packaging into methods	Tutorial	Refer to course web site
	Compatibility and Conversions; Packaging into methods	Workshop	
	Lab 6	Laboratory	
8	Geometry for graphics	Lecture	Refer to course web site
	Geometry for graphics	Tutorial	Refer to course web site
	Geometry for graphics	Workshop	
	Lab 7	Laboratory	
9	Arrays; About programming languages	Lecture	Refer to course web site
	Arrays; About programming languages	Tutorial	Refer to course web site
	Arrays; About programming languages	Workshop	
	Lab 8	Laboratory	
10	Threads; Bolean algebra; searching	Lecture	Refer to course web site
	Threads; Bolean algebra; searching	Tutorial	Refer to course web site
	Threads; Bolean algebra; searching	Workshop	
	Lab 9	Laboratory	
11	Sorting, revision	Lecture	Refer to course web site
	Sorting, revision	Tutorial	Refer to course web site
	Sorting, revision	Workshop	
	Lab 10	Laboratory	
12	Project Demonstrations	Lecture	
	Project Demonstrations	Tutorial	

Assessment

This section sets out the assessment requirements for this course.

Summary of Assessment

Item	Assessment Task	Weighting	Relevant Learning Outcomes	Due Date
1	quiz 1	3%	1,2,3,4,5,6,7,8,9,10,11,12,13	3
2	quiz 2	4%	1,2,3,4,5,6,7,8,9,10,11,12,13	5
3	quiz 3	5%	1,2,3,4,5,6,7,8,9,10,11,12,13	7
4	quiz 4	6%	1,2,3,4,5,6,7,8,9,10,11,12,13	9
5	quiz 5	7%	1,2,3,4,5,6,7,8,9,10,11,12,13	11
6	Assessed lab 1 (exam conditions)	10%	1,2,3,4,5,6,7,8,9,10,11,12,13	4
7	Assessed lab 2 (exam conditions)	15%	1,2,3,4,5,6,7,8,9,10,11,12,13	7
8	Assessed lab 3 (exam conditions)	15%	1,2,3,4,5,6,7,8,9,10,11,12,13	10
9	Project	35%	1,2,3,4,5,6,7,8,9,10,11,12,13	12

Assessment Details

All assessment in this course is individual assessment.

The quizzes motivate and assess your theoretical understanding of the conceptual material delivered in lectures, from personal study, and from laboratory experience.

Due Day and Time: In lectures in weeks 3 (3 marks), 5 (4 marks), 7 (5 marks), 10 (6 marks), 12 (7 marks).

The assessed laboratories allow the application of the theoretical knowledge developed in lectures.

Weightings: Week 5 10%, Weeks 8 & 11 15%

The project offers the opportunity to explore a larger and more interesting real world problem, and is the capstone assessment item for the course. Further information will be given on the course website.

Submission and Return of Assessment Items

Normally you will be able to collect your assignments in class within fourteen [14] days of the due date for submission of the assignment.

Retention of Originals

You must be able to produce a copy of all work submitted if so requested. Copies should be retained until after the release of final results for the course.

Extensions

To apply for an extension of time for an assignment, you must submit an Application for Extension of Assignment form to your teacher at least 24 hours before the date the assignment is due. Grounds for extensions are usually: serious illness, accident, disability, bereavement or other compassionate circumstances and must be able to be substantiated with relevant documentation [e.g. Griffith College Medical Certificate]. Please refer to the Griffith College website - Policy Library - for guidelines regarding extensions and deferred assessment.

Assessment Feedback

Feedback for laboratory exercises will be immediate and in the laboratory session in which they are submitted. Feedback for quizzes will be given the subsequent week in lectures and via the course website.

Feedback for the in-laboratory demonstration component of the project will be immediate. Feedback for the electronically submitted written component of the project will be returned electronically via email and/or the course web site within two weeks.

Generic Skills

Griffith College aims to develop graduates who have an open and critical approach to learning and a capacity for lifelong learning. Through engagement in their studies, students are provided with opportunities to begin the development of these and other generic skills.

Studies in this course will give you opportunities to begin to develop the following skills:

Generic Skills	Taught	Practised	Assessed
Written Communication	Yes	Yes	Yes
Oral Communication		Yes	
Information Literacy	Yes	Yes	Yes
Secondary Research			
Critical and Innovative Thinking	Yes	Yes	Yes
Academic Integrity	Yes	Yes	Yes
Self Directed Learning		Yes	
Team Work		Yes	
Cultural Intelligence		Yes	
English Language Proficiency		Yes	

Additional Course Generic Skills

Additional Course Information

All course material available on the course website:
jasperfly.com

Teacher and Course Evaluations
Students enjoy their journey of acquiring fundamental knowledge and skills in programming. They appreciate the benefits of regular assessment as these are intended to help them reach critical learning milestones at a comfortable pace. In response to student suggestion “to get more guided help on the topic”, teaching and peer support has been augmented with online tools to build their knowledge and skills rapidly. After successfully completing this course, the majority of students feel more confident with programming and therefore perform better in Software Development (1802ICT).

Your feedback is respected and valued by your lecturers and tutors. You are encouraged to provide your thoughts on the course and teaching, both positive and critical, directly to your lecturer and tutor or by completing course and lecturer evaluations via Griffith College’s online evaluation tool whenever these are available.

Academic Integrity

Griffith College is committed to maintaining high academic standards to protect the value of its qualifications. Academic integrity means acting with the values of honesty, trust, fairness, respect and responsibility in learning, teaching and research. It is important for students, teachers, researchers and all staff to act in an honest way, be responsible for their actions, and show fairness in every part of their work. Academic integrity is important for an individual's and the College's reputation.

All staff and students of the College are responsible for academic integrity. As a student, you are expected to conduct your studies honestly, ethically and in accordance with accepted standards of academic conduct. Any form of academic conduct that is contrary to these standards is considered a breach of academic integrity and is unacceptable.

Some students deliberately breach academic integrity standards with intent to deceive. This conscious, pre-meditated form of cheating is considered to be one of the most serious forms of fraudulent academic behaviour, for which the College has zero tolerance and for which penalties, including exclusion from the College, will be applied.

However, Griffith College also recognises many students breach academic integrity standards without intent to deceive. In these cases, students may be required to undertake additional educational activities to remediate their behaviour and may also be provided appropriate advice by academic staff.

As you undertake your studies at Griffith College, your lecturers, tutors and academic advisors will provide you with guidance to understand and maintain academic integrity; however, it is also your responsibility to seek out guidance if and when you are unsure about appropriate academic conduct.

In the case of any allegation of academic misconduct made against a student he or she may request the guidance and support of a Griffith College Student Learning Advisor or Student Counsellor.

Please ensure that you are familiar with the Griffith College Academic Integrity Policy; this policy provides an overview of some of the behaviours that are considered breaches of academic integrity, as well as the penalties and processes involved when a breach is identified.

For further information please refer to the Academic Integrity Policy on the Griffith College website – Policy Library.

Risk Assessment Statement

There are no out of the ordinary risks associated with this course.
