Course Code:	**1005ICT**
Course Name:	**Programming II**
Semester:	**Semester 1, 2016**
Program:	Diploma of Information Technology
Credit Points:	10
Course Coordinator:	Seyedali Mirjalili
Document modified:	11 Dec 2015 12:32:14

Teaching Team

Your lecturer/tutor can be contacted via the email system on the portal.	
Name	**Email**
Seyedali Mirjalili	ali.mirjalili@staff.griffithcollege.edu.au

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 also note: 1005ICT is a prerequisite for courses: 2002ICT, 2501ICT, 2506ICT, 3512ICT, 3611ICT and 2503ICT.

This means that you need to achieve a Pass or above to progress to these courses.

- 1001ICT - Programming 1

Brief Course Description

Programming II is a 10 credit point course within the Diploma of Information Technology. The course is situated within the second 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.

This course builds on 1001ICT Programming I, focusing on fundamental concepts underlying the object-oriented approach to programming by introducing the widely-used Java programming language. By concentrating on aspects of Java that best demonstrates object-oriented principles and good practice, students gain a solid basis for further study of the Java language, and of object-oriented software development. The course introduces students to fundamental problem-solving, software design and programming techniques and their application to a range of selected problems.

In the object-oriented view of software, programs are considered to be collections of objects that interact by sending messages to one another and reacting to the responses to those messages. These concepts are at the forefront of modern software development and are used in a most commonly-used programming languages.

This course covers the fundamental concepts associated with object-oriented programming (objects, classes, inheritance and polymorphism). Students will learn to apply and extend their knowledge of programming concepts within an object-oriented programming context, the Java programming language.

The concepts will be transferrable to other courses and languages, such as JavaScript, ActionScript, C++, C# and Objective-C.

Rationale

IT professionals need to possess practical understanding and skill in programming concepts including the analysis, design and implementation of software. This course is designed to extend problem solving and programming skills introduced in 1001ICT Programming I.

Programming II aims to provide a comprehensive introduction to object-oriented programming using the Java programming language, and provide an introduction to systematic techniques for developing, testing and documenting high-quality software in Java.

In the object-oriented view of software, programs are considered to be collections of objects that interact by sending messages to one another and reacting to the responses to those messages. These concepts are at the forefront of modern software development and are used in a number of programming languages.

After a revision of procedural programming (basic statements and methods), this course takes an “objects first” approach to teaching and covers some of the fundamental concepts associated with object-oriented programming (objects, classes, inheritance and polymorphism). Students will learn to apply and extend their knowledge of programming concepts within an object-oriented programming context.

Aims

This course aims to provide the required background for all further studies in software engineering and computer science, and valuable background for further studies in other areas of information technology or experience in the IT industry. The course also aims to develop your critical evaluation, analysis and problem solving skills.

Learning Outcomes

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

1 Understand the concepts behind object-oriented programming;
2 Understand and create Unified Modelling Language class diagrams;
3 Understand the syntax and semantics of the Java programming language;
4 Understand and use the Java class library;
5 Design and apply reusable Java software components;
6 Effectively use Java to solve simple programming problems;
7 Use basic software engineering skills for designing, implementing, testing and documenting small-sized programs using object-oriented programming.

Texts and Supporting Materials

Prescribed Text:

- Andrew Rock, 1005ICT Introduction to Object Orientated Programming Lecture Notes, 2014 Edition, Griffith College, 2014.
- John Lewis, Peter DePasquale, Joseph Chase, Java Foundations: Introduction to Program Design and Data Structures, 2nd Edition, 2011

Recommended Reading:

- Oracle Java API Documentation. Provided online.
  Ken Arnold, James Gosling and David Holmes, The Java Programming Language, Third Edition, Sun Press/Addison-Wesley, 2001.
- Kathy Sierra and Bert Bates, Head start Java, 2nd Edition, O\\\\\\\\\\\\\\\'Reilly Media, 2006.
- Brett D. McLaughlin, Gary Pollice, Dave West, Head first Object-Orientated Design and Analysis, 1st Edition, O\\\\\\\\\\\\\\\'Reilly Media, 2006.

Organisation and Teaching Strategies

Throughout the semester, the techniques introduced will be illustrated by relevant programming examples from a variety of domains.

The course has one two-hour lecture, one two-hour laboratory class each week. The course topics are introduced during the lecture time. It is expected that you will expand your understanding of this material by reviewing the additional course material provided on the portal. The additional material is provided in the form of written course notes, example programs and pre-recorded lectures available on the course website. The new material is explained, illustrated, and reinforced during lectures, with added embedded exercises. This material is then further reinforced and extended in laboratory classes, where you will complete assessed laboratory exercises, receive advice and feedback on programming projects, and are able to request clarification on all aspects of the course.

Extensive teaching materials, including lecture notes, recorded lectures, example programs, tutorials and selected reference material is available on the course web site. The recommended texts are used as a reference for the programming environment, to reinforce and extend material covered in lectures, as a source of additional programming examples, and as a reference to the Java programming language.

Weekly labs will allow an opportunity to practice the topics covered in lectures, and in weeks 4, 7, and 11 these areas will be assessed. Students will use a pairwise programming approach during non-assessed labs.

Class Contact Summary

Attendance

Your attendance in class will be marked each hour. To receive full attendance, you must be present in the classroom no later than 15 minutes after the start of the class. Therefore, you are encouraged to attend and participate in all classes throughout the semester.

You are required to attend the two lectures each week and to attend and participate in your enrolled laboratory class each week. Failure to attend and participate in required classes may be taken into consideration by the teaching team if you request out-of-hours assistance, or special consideration.

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, but is highly recommended that you purchase a printed set from the book shop and bring them to each class so that extra notes can be added.

Independent Learning

You are expected to spend 1 hour per credit point per week on course related activities which include attending lectures, tutorials, workshops, reading the recommended texts / lecture notes, research and revision. It is not expected that the total workload for the course will exceed the normal expectations of a 10 CP course. This is on average 10 hours per week. Even though you may feel that less time is required at the beginning, it is very easy to get behind and become overwhelmed. So please spend an appropriate amount of time each week. Use the following as a guide:

- **Lecture Preparation** (watching associated lecture video, reading notes, skimming textbook, following tutorials, experimenting with examples) -**3.5 hours**
- **Laboratory** (in class) -**2 hours**
- **Lecture** (in class)- **2 hours**
- **Preparation for lab activities-2 hours**
- **Learning something new about programming -30 minutes.**

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

The course will roughly follow the sequence of topics in the prescribed text. Exercises in laboratory classes will review material covered in the preceding weeks.

The content of the course consists of the following main topics:

1. Course overview and introduction
2. Kick-start Java
3. Programming in the Large
4. Case Study: Class java.lang.Math
5. Basic Java Data Types
6. Case Study: Class java.lang.String
7. Case Study: java.util.Scanner
8. Essential Object Oriented Programming
9. Case Study: Number Classes
10. Errors and Exceptions
11. Java Language Goodies
12. The Software Development Process
13. UML Class Diagrams
14. 2D Graphics In a Window
15. Advanced Data Structures
16. Generics
17. Interfaces
18. The Collections Framework
19. Inner Classes
20. GUI Components and Events

Weekly Teaching Schedule

Week	Topic	Activity	Readings
1	Introduction to 1005ICT and Java	Lecture	Lecture notes section 1, 2
	Working with the JAVA Tools	Workshop	
2	3. Programming In the Large 4. Case Study: Class java.lang.Math	Lecture	Lecture notes section 3, 4
	Intro to JAVA Classes	Tutorial	
3	5. Basic Java Data Types 6. Case Study: Class java.lang.String 7. Case Study: java.util.Scanner	Lecture	Lecture notes section 5, 6, 7
	Lab Activity: Dynamic Classes	Tutorial	
4	8. Essential Object Oriented Programming 9. Case Study: Number Classes	Lecture	Lecture notes section 8, 9
	Assessed Lab 1	Assessment	
5	10. Errors and Exceptions 11. Java Language Goodies	Lecture	Lecture notes section 10, 11
	Lab Activity: Using the API (Scanner with STDIO)	Tutorial	
6	12. The Software Development Process 13. UML Class Diagrams	Lecture	Lecture notes section 12, 13
	Lab Activity: UML to Code	Tutorial	
7	14. 2D Graphics In a Window 15. Advanced Data Structures	Lecture	Lecture notes section 14, 15
8	16. Generics	Lecture	Lecture notes section 16
	Lab Activity: 2D Graphics	Tutorial	
	Assessed Lab 2	Assessment	
9	17. Interfaces 18. The Collections Framework	Lecture	Lecture notes section 17, 18
	Lab Activity: Animation	Tutorial	
10	19. Inner Classes 20. GUI Components and Events	Lecture	Lecture notes section 19, 20
	Lab Activity: Collection Framework	Tutorial	
11	Designing Software	Lecture	
	Assessed Lab 3	Tutorial	
12	Documenting and design	Lecture	
	Project	Workshop	
13	Finalizing the Project	Lecture	
	Project	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	Assessed Lab 1	10%	1,2,3,4,5,6,7	4
2	Assessed Lab 2	15%	1,2,3,4,5,6,7	8
3	Assessed Lab 3	15%	1,2,3,4,5,6,7	11
4	Quiz 1	3%	1,2,3,4,5,6	3
5	Quiz 2	4%	1,2,3,4,5,6	5
6	Quiz 3	5%	1,2,3,4,5,6	8
7	Quiz 4	6%	1,2,3,4,5,6	10

8	Quiz 5	7%	1,2,3,4,5,6	12
9	Project	35%	1,2,3,4,5,6,7	13

Assessment Details

All assessment in this course is individual assessment.

Quizzes (5)

The quizzes motivate and assess the student\\\\\\\\\\\\\\\'s 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).

Total marks: 25

Further information will be given on the course website, and a practice quiz will be run in a workshop.

Each quiz will take about 10 to 20 minutes of lecture time.

The questions will be multiple choice, and test theory knowledge, the ability to read and understand program code and what it does, and detect possible defects.

Assessed Laboratories (3)

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

Further information will be given on the course website.

Project

The project offers the opportunity to explore larger and more interesting real world problems, and is the capstone assessment item for the course.

Total Marks: 35

Weighting: 35%

Further information will be given on the course website.

Marking will be mainly based on meeting the techical requirements of the project exercises, with a small weighting given to documentation and presentation.

Other Information

You are responsible for maintaining copies of assessment drafts prior to submission (including electronic backups). No extensions or special consideration will be given if you are unable to submit an assessment because of data loss or corruption. No extensions or special consideration will be given if you are unaware of assessment deadlines.

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

Marks awarded for assessment items will also be available on the on-line grades system on the Student Website within fourteen [14] days of the due date. You may arrange an appointment during the designated consultation time to discuss assessment in more detail.

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	
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			
Cultural Intelligence		Yes	
English Language Proficiency		Yes	

Additional Course Generic Skills

Specific Skills	Taught	Practised	Assessed
Abstract thinking	Yes	Yes	Yes

Additional Course Information

All course material is available on the course website.

Teacher and Course Evaluation

Students find the opportunity to build upon the concepts taught in Programming 1 to design and build solutions to real problems both exciting and challenging. Past studentsâ€™ feedback has been instrumental in refining course delivery and assessment structure in terms of designing preparation activities a week before the lab, as well as restructuring assessment so that now there are only four labs instead of ten. These changes facilitated studentsâ€™ active involvement in the weekly lessons, including practicing their skills on a regular basis.

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 Collegeâ€™s online evaluation tool whenever these are available.

Pass Conceded Grades

Please note that there are no PC (Pass Conceded) grades awarded for this course.

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.

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.

Note: For all Diploma level programs, Griffith College acknowledges content derived from Griffith University.