

INFS1609

Fundamentals of business Programming

Course Outline Semester 2, 2017

Course-Specific Information

The Business School expects that you are familiar with the contents of this course outline. You must also be familiar with the Course Outlines Policies webpage which contains key information on:

- Program Learning Goals and Outcomes
- Academic Integrity and Plagiarism
- Student Responsibilities and Conduct
- Special Consideration
- Student Support and Resources

This webpage can be found on the Business School website:

<https://www.business.unsw.edu.au/degrees-courses/course-outlines/policies>

Table of Contents

COURSE-SPECIFIC INFORMATION	1
1 STAFF CONTACT DETAILS	1
2 COURSE DETAILS	1
2.1 Teaching Times and Locations	1
2.2 Units of Credit	1
2.3 Summary of Course	1
2.4 Course Aims and Relationship to Other Courses	2
2.5 Student Learning Outcomes	2
3 LEARNING AND TEACHING ACTIVITIES	3
3.1 Approach to Learning and Teaching in the Course	3
3.2 Learning Activities and Teaching Strategies	4
4 ASSESSMENT	5
4.1 Formal Requirements	5
4.2 Assessment Details	5
4.3 Special Consideration, Late Submission and Penalties	7
4.4 Protocol for viewing final exam scripts	7
5 COURSE RESOURCES	7
6 COURSE EVALUATION AND DEVELOPMENT	7
7 COURSE SCHEDULE	8

COURSE-SPECIFIC INFORMATION

1 STAFF CONTACT DETAILS

Position	Name	Email	Room
Lecturer-in-charge	Yenni Tim	yenni.tim@unsw.edu.au	2102, Level 2, Quad Building West Wing

LIC Consultation Time – **Wednesday, 13:00 – 15:00** (or by appointment)

The best way to contact your lecturer or tutor is via email or to see them during their consultation times. Please use your **UNSW email account** to contact staff members. The subject of your email should begin with the course code (i.e., INFS1609).

To protect student privacy, staff members will not answer queries or disclose information relating to assessments of students if a non-UNSW email account is used. Please note that formal notices and correspondence regarding the course will only be sent to your UNSW email account.

Please contact the School Office (QUAD2091) should you not be able to contact either the Lecturer or your tutor. Alternatively, if you need to contact the school urgently, ring +61 (2) 9385-5320 or email: istm@unsw.edu.au.

2 COURSE DETAILS

2.1 Teaching Times and Locations

Lectures start in Week 1 (to Week 12)

Time: Thursday, 17:00 – 18:00

Location: Mathews Theatre A (K-D23-201)

Tutorials start in Week 2 (to Week 13). A full list of tutorials, times and tutors will be posted on the Course Website. For latest information about the lecture and tutorial locations, see: <http://timetable.unsw.edu.au/2017/INFS1609.html>.

You must attend the workshop in which you are enrolled. If you are unable to attend your allocated workshop due to illness or misadventure, you must contact your LiC as soon as possible.

Peer Assisted Study Sessions (PASS)

Pass sessions run from Week 3 to Week 13. Timetables for the PASS groups will be posted on the Course Website.

2.2 Units of Credit

The course is worth 6 units of credit.

2.3 Summary of Course

This is a foundational (Level 1) Information Systems (IS) course that introduces students to application programming. The course provides a first step towards learning the

principles of object-oriented programming through the Java programming language. Programming refers to the development of software, which is also called a program. Essentially, software contains the instructions that tell computerised devices what to do. In lectures, students will be introduced to the theoretical component of the course, learning fundamental programming concepts. During weekly workshop tutorials, students will engage in the practical component of the course, learning how to write code using the NetBeans Integrated Development Environment.

The topics that are covered in INFS1609 introduce students to the fundamentals of Java programming. This begins with an overview of data types and methods before introducing students to small problem-solving exercises that require the use of conditional statements, loops and arrays (including multi-dimensional arrays and array lists). Students are then introduced to the topics of modular programming, testing and debugging (using JUNIT). Finally, having gained a general understanding of these concepts, students further explore the principles of object-oriented programming, including objects, classes, abstraction, polymorphism, inheritance and encapsulation.

2.4 Course Aims and Relationship to Other Courses

This course is offered as one of the cornerstone courses in the discipline of Information Systems. This course aims to develop students' programming and problem solving abilities in preparation for the next-stage courses (such as INFS2605 Intermediate Business Programming, INFS3634 Mobile Application Development and INFS3605 IS Innovation and Transformation). This course also aims to develop students' ability to work individually in solving problems through the application of programming concepts to design. Overall, this course aims to provide students with understanding and skills that are essential in careers such as project managers, business analysts, system analysts, designers and developers.

2.5 Student Learning Outcomes

The Course Learning Outcomes are what you should be able to DO by the end of this course if you participate fully in learning activities and successfully complete the assessment items. On successful completion of this course, you should:

1. Demonstrate ability to interpret and write reliable, well-structured, and well-documented software programs.
2. Describe and apply the principles of object-oriented programming in developing software programs.
3. Demonstrate ability to effectively use an Integrated Development Environment for software development.
4. Develop problem-solving skills in designing and implementing programming solutions.
5. Examine and debug basic software programs.
6. Comprehend and apply software testing methods.

The Learning Outcomes in this course also help you to achieve some of the overall Program Learning Goals and Outcomes for all undergraduate students in the Business School. Program Learning Goals are what we want you to BE or HAVE by the time you successfully complete your degree (e.g. 'be an effective team player'). You demonstrate this by achieving specific Program Learning Outcomes – what you are able to DO by the end of your degree (e.g. 'participate collaboratively and responsibly in teams').

For more information on Program Learning Goals and Outcomes, see the School's Course Outlines Policies webpage available at <https://www.business.unsw.edu.au/degrees-courses/course-outlines/policies>.

The following table shows how your Course Learning Outcomes relate to the overall Program Learning Goals and Outcomes, and indicates where these are assessed (they may also be developed in tutorials and other activities):

<i>This course helps you to achieve the following learning goals for all Business undergraduate students:</i>		<i>On successful completion of the course, you should be able to:</i>	<i>This learning outcome will be assessed in the following items:</i>
1	Knowledge	Demonstrate ability to interpret and write reliable, well-structured, and well-documented software programs. Describe and apply the principles of object-oriented programming in developing software programs. Demonstrate ability to effectively use an Integrated Development Environment for software development. Comprehend and apply software testing methods.	<ul style="list-style-type: none"> • Tutorial Problems • Quizzes 1 & 2 • Final Exam
2	Critical thinking and problem solving	Develop problem-solving skills in designing and implementing programming solutions. Examine and debug basic software programs.	<ul style="list-style-type: none"> • Tutorial Problems • Quizzes 1 & 2 • Final Exam
3a	Written communication	Describe and apply the principles of object-oriented programming in developing software programs.	<ul style="list-style-type: none"> • Tutorial Problems • Quizzes 1 & 2 • Final Exam
3b	Oral communication	Not specifically addressed in this course.	
4	Teamwork	Not specifically addressed in this course.	
5a	Ethical, social and environmental responsibility	Not specifically addressed in this course.	
5b	Social and cultural awareness	Not specifically addressed in this course.	

3 LEARNING AND TEACHING ACTIVITIES

3.1 Approach to Learning and Teaching in the Course

This course introduces you to the foundations of the programming discipline, which underlies most technical subjects such as software and mobile applications design, data management, and algorithms. The course provides a first step towards learning the

principles of object-oriented design and programming using Java programming language. In addition to developing programming skills, the focus of this course is also on self-directed learning and problem solving. Lectures, tutorials, textbooks, quizzes, exams and other resources are all provided to help this process.

We will cover a lot of material in INFS1609, so it is vital that you study from Week 1. Essentially, this means that you should read the course materials and prepare for your workshops. The course team will facilitate your learning by providing the guidance as to what you need to study, and working with you on problems you may encounter. It is, however, your responsibility to make a concerted and timely effort to study. If you make this effort, you will find the material interesting, the course worthwhile and the interaction with your fellow students stimulating. You should also do well.

3.2 Learning Activities and Teaching Strategies

The course involves three key components – lectures, tutorials, and your private study.

Each lecture will outline the main concepts and methods for this course. Each week, the LIC will begin by reviewing and clarifying material previously covered. The LIC will then introduce a new topic, highlight relevant study material and present students with programming examples. On occasion, the LIC will use the lecture time to pose questions to students and hold class discussions on topics covered. The relevant study material and programming exercises, to be read and completed in your own time, provide more details about the topics introduced in the lecture. It is expected that you will spend approximately 10 hours per week studying for this course. This time should be made up of reading, revision, working on exercises and problems, and attending classes (lectures and tutorials). In periods where you need to complete quizzes or prepare for examinations, the workload may be greater.

Tutorials will be used to reinforce and apply concepts covered in lectures and study materials. Tutorials are the key element of the course. Tutorials for INFS1609 will be run as sandboxing workshops. These workshops provide an intellectually stimulating hands-on environment for students to experiment and design simple algorithms (using Java programming language) to solve problems. Over the semester, you will engage in a variety of different problem-solving scenarios that build in complexity and that call for different combinations of knowledge and skills. Being prepared for your workshops is therefore essential. Student should routinely check what materials they are expected to read/complete prior to each sandboxing session. This includes completing any activities you have been asked to do in preparation for your next workshop as well as reviewing your lecture notes from your previous lecture. The sandboxing workshop will also give you the opportunity to discuss your work with fellow students and tutor, and hence gain an indication of your own progress. Students should also use their workshop time to ask questions for clarifications on the material covered in class as well as their study material.

Your private study is the most important component of this course. The textbook contains self-assessment exercises to help you. The self-assessment exercises are designed to test your understanding of the topic at hand and include review questions, application questions and discussion questions of varying difficulty. The course site on Moodle will provide you with access to additional materials.

Peer Assisted Study Sessions (PASS). PASS are free, weekly, out-of-class study sessions available to all students enrolled in this course. They are facilitated by a leader (or leaders), who is (are) student(s) who have previously enrolled for and successfully completed the course. Attending PASS regularly can help you to:

- Deepen your understanding of the course content
- Develop skills for independent university study
- Make friends
- Feel more confident in your studies

PASS sessions begin in week 3. Timetables for the PASS groups will be made available on the Moodle website. There is no need to register. It is recommended that you attend the same group regularly but there is no obligation. You can even attend more than one PASS group a week if you like. You can also choose to attend some weeks but not others.

4 ASSESSMENT

4.1 Formal Requirements

To receive a pass grade in this course, you must meet ALL of the following criteria:

- Achieve a composite mark of at least 50.
- Attend at least 80% of all scheduled classes.
- Attain a satisfactory performance in each component of the course. A mark of 45% or higher is normally regarded as satisfactory.
- Attain a mark of at least 45% in the final exam.

The School reserves the right to scale final marks to a mean of 60%.

4.2 Assessment Details

Assessment Task	Weighting	Length	Due Date
Tutorial Participation	20%	See below	Tutorials, Weeks 2 – 13
Quizzes 1 & 2	30%	See below	Week 7 and Week 12 lecture time
Final Exam	50%	2 hours	University Exam Period
Total	100%		

Tutorial Preparation and Participation (20%)

Tutorials will be run as weekly sandboxing workshops from Week 2 to Week 13 (with the exception of Week 10, which has no workshops). A sandboxing workshop provides an intellectually stimulating, hands-on environment where students will be engaged in problem solving and design. The role of the sandboxing workshops is to help build your understanding and problem solving skill through the application of what you have learnt (i.e., fundamentals of programming) to solving simple, simulated business problems. This year, tutorial preparation and participation (P&P) has a weighting of 20% in total.

Please note: All students are expected to adhere to **their allocated workshop times**. Students are required to prepare for each workshop and the workshop will require your **full participation**. Marks will be given for students who have prepared (i.e., completed any necessary self-study and preparation work), are on-time for the workshops and

actively participate in the sandboxing activities. Active participation includes, but is not limited to: providing programming solutions for exercises; engaging in discussions; asking and answering questions; and taking notes. Students who are not prepared for a workshop, are late for the workshop, and/or are not fully engaged during the workshop itself (e.g. occupied with social networking, surfing the web, checking mail, etc.) may not be awarded an assessment mark.

A tutorial preparation and participation grading guide will be uploaded on Moodle website to help you understand the grading criteria. It is strongly recommended that you understand the grading rubric as this will help you in your preparation for, and participation in, tutorials each week. Expectations for workshop preparation and participation will also be discussed in your first lecture.

Students will also be required to sign an attendance sheet each week. It is your responsibility to ensure that you arrive on time and sign the attendance sheet. Without a signature or failure to sign on the attendance roll **before the end of the 10th minute from the start time of the workshop**, you will be marked as ABSENT even if you have attended the workshop or arrived on time.

You must attend the workshop in which you are enrolled. If you are unable to attend your allocated workshop due to illness or misadventure, you must contact your LiC as soon as possible.

As stated in Section 1, please note that only your UNSW email account should be used for formal notices and correspondence regarding the course. Always sign your email with your name and student number. The subject of your e-mail should begin with the course code (i.e. INFS1609).

Quizzes (30%)

There are two quizzes for this course. Each quiz will incorporate a set of multiple choice questions. Quiz 1 will be administered in Week 7 lecture time and has a weighting of 10%; Quiz 2 will be administered in Week 12 lecture time and has a weighting of 20%. Therefore, quizzes for INFS1609 have a weighting of 30% in total. The quizzes will be in line with the topics covered in the lectures, tutorials, and study material. However, students will need to engage in their own study and coding exercises to do well in the quizzes.

IMPORTANT: Both quizzes will be administered in an **examination condition**. Students will be required to arrive at the lecture venue **not less than 5 minutes before the lecture time** in Week 7 and Week 12 and display the UNSW student identification card for the duration of the quizzes. More specific details for each quiz (e.g., number of questions, length and sample questions) and additional rules governing the conduct of the quizzes will be uploaded on Moodle website.

Final Examination (50%)

A formal, closed-book, written examination will take place during the University Exam Period. The examination time will be 2 hours. The examination is worth 50% of the total marks for this course. You must plan to be available for the full examination period to attend the final exam. In addition, you should also ensure that you will be available for a supplementary examination in the event of illness or misadventure. All material covered in lectures, tutorials, exercises, and set readings is examinable. All exams are conducted in accordance with the UNSW Rules for the Conduct of Examinations and it is your responsibility to be familiar with these rules.

4.3 Special Consideration, Late Submission and Penalties

For information on Special Consideration please refer to the Business School's [Course Outlines Policies webpage](#).

4.4 Protocol for viewing final exam scripts

The School of Information Systems and Technology Management (ISTM) has set a protocol under which students may view their final exam script. ISTM exam script viewing day is usually a day after the official release of results. Details will be posted on both the school website and on your course Moodle.

Quality Assurance

The Business School is actively monitoring student learning and quality of the student experience in all its programs. A random selection of completed assessment tasks may be used for quality assurance, such as to determine the extent to which program learning goals are being achieved. The information is required for accreditation purposes, and aggregated findings will be used to inform changes aimed at improving the quality of Business School programs. All material used for such processes will be treated as confidential.

5 COURSE RESOURCES

The website for this course is on Moodle at:
<http://moodle.telt.unsw.edu.au>

The **required** textbook for this course is:

Intro to Java Programming, Brief Version
by Liang Y Daniel
10th Edition, Pearson Education, 2014
ISBN: 978-0133592207

6 COURSE EVALUATION AND DEVELOPMENT

Each year feedback is sought from students and other stakeholders about the courses offered in the School and continual improvements are made based on this feedback. UNSW's myExperience survey is one of the ways in which student evaluative feedback is gathered. In this course, we will seek your feedback through end of semester myExperience responses.

7 COURSE SCHEDULE

COURSE SCHEDULE			
Week	Topic	References	Other activities / assessment
Week 1 24 July	Introduction to Programming	Chapter 1	<i>NO TUTORIALS</i>
Week 2 31 July	Elementary Programming 1	Chapter 2	Tutorial 1
Week 3 7 August	Elementary Programming 2	Chapter 2	Tutorial 2
Week 4 14 August	Selections	Chapter 3	Tutorial 3
Week 5 21 August	Loops	Chapter 4	Tutorial 4
Week 6 28 August	Methods	Chapter 5	Tutorial 5
Week 7 4 September	Single-dimensional Arrays	Chapter 6	Tutorial 6 and Quiz 1
Week 8 11 September	Single and multi-dimensional Arrays	Chapter 7	Tutorial 7
Week 9 18 September	Thinking in Objects	Chapter 10	Tutorial 8
Mid-semester break: 23 September – 2 October inclusive (2 Oct = Labour Day Public Holiday)			
Week 10 3 October	Thinking in Objects	Chapter 10	<i>NO TUTORIALS</i>
Week 11 9 October	Abstraction, Polymorphism, Inheritance and Encapsulation	Chapter 11	Tutorial 9
Week 12 16 October	Abstraction, Polymorphism, Inheritance and Encapsulation	Chapter 11	Tutorial 10 and Quiz 2
Week 13 23 October	<i>Revision Lecture</i>		Tutorial 11