INFS1609
FUNDAMENTALS OF BUSINESS
PROGRAMMING

AND

INFS2609
PROGRAMMING FOR BUSINESS

Course Outline
Semester 2, 2015

Part A: Course-Specific Information

Please consult Part B for key information on Business School policies (including those on plagiarism and special consideration), student responsibilities and student support services.
## Table of Contents

**PART A: COURSE-SPECIFIC INFORMATION**

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. STAFF CONTACT DETAILS</td>
<td>1</td>
</tr>
<tr>
<td>2. COURSE DETAILS</td>
<td>1</td>
</tr>
<tr>
<td>2.1 Teaching Times and Locations</td>
<td>1</td>
</tr>
<tr>
<td>2.2 Peer Assisted Study Sessions (PASS)</td>
<td>1</td>
</tr>
<tr>
<td>2.3 Units of Credit</td>
<td>1</td>
</tr>
<tr>
<td>2.4 Summary of Course</td>
<td>1</td>
</tr>
<tr>
<td>2.5 Course Aims and Relationship to Other Courses</td>
<td>2</td>
</tr>
<tr>
<td>2.6 Student Learning Outcomes</td>
<td>2</td>
</tr>
<tr>
<td>3. LEARNING AND TEACHING ACTIVITIES</td>
<td>4</td>
</tr>
<tr>
<td>3.1 Approach to Learning and Teaching in the Course</td>
<td>4</td>
</tr>
<tr>
<td>3.2 Learning Activities and Teaching Strategies</td>
<td>4</td>
</tr>
<tr>
<td>4. ASSESSMENT</td>
<td>5</td>
</tr>
<tr>
<td>4.1 Formal Requirements</td>
<td>5</td>
</tr>
<tr>
<td>4.2 Assessment Details</td>
<td>6</td>
</tr>
<tr>
<td>4.3 Assignment Submission Procedure</td>
<td>7</td>
</tr>
<tr>
<td>4.4 Late Submission</td>
<td>7</td>
</tr>
<tr>
<td>5. COURSE RESOURCES</td>
<td>8</td>
</tr>
<tr>
<td>6. COURSE EVALUATION AND DEVELOPMENT</td>
<td>8</td>
</tr>
<tr>
<td>7. COURSE SCHEDULE</td>
<td>9</td>
</tr>
</tbody>
</table>
PART A: COURSE-SPECIFIC INFORMATION

1 STAFF CONTACT DETAILS

<table>
<thead>
<tr>
<th>Position</th>
<th>Name</th>
<th>Email</th>
<th>Room</th>
<th>Phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lecturer-in-charge</td>
<td>Dr Ben Choi</td>
<td><a href="mailto:chun.choi@unsw.edu.au">chun.choi@unsw.edu.au</a></td>
<td>Room 2113, Quadrangle</td>
<td>9385 9843</td>
</tr>
</tbody>
</table>

LIC Consultation Time: Monday, 14:00 – 16:00 (by appointment only)

The preferred method of contacting your lecturer or tutor is through email. Your UNSW e-mail account should be used for formal notices and correspondence regarding the course. For security reasons, please avoid using e-mails from anonymous accounts, such as Yahoo, Hotmail, and Gmail. Always start the subject line of emails with INFS1609/INFS2609 and sign the email with your full name and student number.

If you need to contact the School urgently you can contact the School Office on 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)
The Time: Wednesday 11:00-12:00
The Location: ColomboThC

Tutorials start in Week 2 (to Week 13).
The Groups and Times are as follows:

Group M10A: Monday 10:00-12:00
Group M12A: Monday 12:00-14:00
Group M11A: Tuesday 11:00-13:00
Group M13A: Tuesday 13:00-15:00

2.2 Peer Assisted Study Sessions (PASS)

PASS sessions run from week 3 to week 13.
Timetables for the PASS groups will be announced later.

2.3 Units of Credit

The course is worth 6 units of credit.

2.4 Summary of Course

This course aims to introduce students to the foundations of the programming discipline. It will involve both a theoretical component (e.g., learning about basic programming concepts like loops, arrays, and functions) as well as a practical component (e.g., implementing simple algorithms in a computer laboratory). The course also provides a first step towards learning the principle of object-oriented design and programming through the use of the Java programming language.
2.5 Course Aims and Relationship to Other Courses

This course covers material that is significant to the discipline of Information Systems. A central aim of this course is to build on students' programming skills in preparation for INFS2605 (Business Application Programming). 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 various concepts and skills that are essential in careers such as project managers, business analysts, systems analysts, designers, and developers.

2.6 Student Learning Outcomes

On successful completion of this course, you should:

1. Be familiar with the concepts of object-orient programming: classes, objects, fields, and methods.
2. Be able to read, write and debug simple programs.
3. Be able to design programs to solve simple tasks using storyboards and flowcharts.
4. Be able to decompose a complex task into simpler functions and methods.
5. Understand and be able to use conditional statements, loops and recursion to make more complex programs.
6. Be able to make interactive programs using events.
7. Be able to use random numbers to make programs which respond non-deterministically.
8. Be able to use lists and arrays to handle sequences of data.
9. Be able to understand basic file input and output operations.

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.

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 the Undergraduate Program Learning Goals and Outcomes, see Part B of the course outline.
Business Undergraduate Program Learning Goals and Outcomes

1. Knowledge: Our graduates will have in-depth disciplinary knowledge applicable in local and global contexts.
   You should be able to select and apply disciplinary knowledge to business situations in a local and global environment.

2. Critical thinking and problem solving: Our graduates will be critical thinkers and effective problem solvers.
   You should be able to identify and research issues in business situations, analyse the issues, and propose appropriate and well-justified solutions.

3. Communication: Our graduates will be effective professional communicators.
   You should be able to:
   a. Prepare written documents that are clear and concise, using appropriate style and presentation for the intended audience, purpose and context, and
   b. Prepare and deliver oral presentations that are clear, focused, well-structured, and delivered in a professional manner.

4. Teamwork: Our graduates will be effective team participants.
   You should be able to participate collaboratively and responsibly in teams, and reflect on your own teamwork, and on the team’s processes and ability to achieve outcomes.

5. Ethical, social and environmental responsibility: Our graduates will have a sound awareness of the ethical, social, cultural and environmental implications of business practice.
   You should be able to:
   a. Identify and assess ethical, environmental and/or sustainability considerations in business decision-making and practice, and
   b. Identify social and cultural implications of business situations.

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):

<table>
<thead>
<tr>
<th>Program Learning Goals and Outcomes</th>
<th>Course Learning Outcomes</th>
<th>Course Assessment Item</th>
</tr>
</thead>
<tbody>
<tr>
<td>This course helps you to achieve the following learning goals for all Business undergraduate students:</td>
<td>On successful completion of the course, you should be able to:</td>
<td>This learning outcome will be assessed in the following items:</td>
</tr>
<tr>
<td>1 Knowledge</td>
<td>Apply programming concepts to design. Explain, apply, and evaluate object-oriented design. Explain and apply testing and debugging.</td>
<td>• Tutorial Problems  • Assignment 1, 2, &amp; 3  • In-tutorial Quizzes  • Exam</td>
</tr>
<tr>
<td>2 Critical thinking and problem solving</td>
<td>Evaluate and provide solutions to small scale problems. Design programs that complete computation tasks.</td>
<td>• Tutorial Problems  • Assignment 1, 2, &amp; 3  • In-tutorial Quizzes  • Exam</td>
</tr>
</tbody>
</table>
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 design, data management, and algorithms. The course provides a first step towards learning the principles of object-oriented design and programming through the use of the Java programming language.

We will cover a lot of material in INFS1609/INFS2609, 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, workshops, 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, highlighting relevant study material and presenting students with programming exercises to be completed before the following week’s tutorial. 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, to be read in your own time, provides more detail 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 assignments or prepare for examinations, the workload may be greater.

Tutorials will be used to reinforce and apply material covered in lectures and study material. Tutorials are an important part of your learning; therefore, being prepared for...
your tutorials is essential. Student should routinely check what material they are expected to read/complete prior to each session. This includes completing any activities you have been asked to do in preparation for your next tutorial as well as reviewing your lecture notes from your previous lecture.

Tutorials also give you the opportunity to discuss your work with fellow students, and hence gain an indication of your own progress. Students should also use their tutorial time to ask questions for clarifications on the material covered in class as well as their study material.

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.

During weeks 3-12, tutorials will: (i) assess students’ knowledge through quizzes; and (ii) assess students through problem exercises that will require students (both individuals and teams) to design and implement appropriate programming solutions.

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 offers free, weekly, out-of-class study sessions available to all students enrolled in INFS1602. They are facilitated by a leader (or leaders) who have previously and successfully completed the course. Attending PASS regularly can help you to:

- Learn in a supportive environment from your peers
- Meet other students and friends
- Reinforce what you learn in lectures and workshops
- Practice problems with experienced leaders
- Learn how to study effectively
- Feel free to ask any questions

PASS sessions begin in week 3. 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.

Timetable of PASS will be available on Moodle.

4 ASSESSMENT

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

- Attain an overall 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 percent 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

<table>
<thead>
<tr>
<th>Assessment Task</th>
<th>Weighting</th>
<th>Length</th>
<th>Due Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tutorial Participation</td>
<td>10%</td>
<td>See below</td>
<td>Tutorials, Weeks 2 – 13</td>
</tr>
<tr>
<td>Quizzes (1-10)</td>
<td>10%</td>
<td>See below</td>
<td>Tutorials, Weeks 3 – 13</td>
</tr>
<tr>
<td>Programming Assignments 1, 2 &amp; 3</td>
<td>30%</td>
<td>See below</td>
<td>Week 5, 9, and 12</td>
</tr>
<tr>
<td>Final Exam</td>
<td>50%</td>
<td>2 hours</td>
<td>University Exam Period</td>
</tr>
</tbody>
</table>

**Tutorial Participation**

Tutorials will be used to reinforce material covered in lectures as well as study material. Each tutorial will involve a number of problems/exercises which relate to a given topic. During weeks 2–13 (with the exception of week 10, which has no tutorial classes), active participation in tutorials is included as part of student assessment for INFS1609/INFS2609. Tutorial participation has a weighting of 10% in total.

Active participation includes, but is not limited to: providing programming solutions for exercises; engaging in tutorial discussions; asking and answering questions; and taking notes. Students will also be required to sign an attendance sheet each week.

Please note: All students are expected to be punctual and to adhere to their allocated tutorial times. Latecomers may not be awarded an assessment mark. Students are required to prepare for each tutorial and the tutorial will require your full participation. Students who are not prepared for a tutorial and/or are not fully engaged during the tutorial itself (e.g. occupied with social networking, surfing the web, checking mail, etc) may not be awarded an assessment mark.

**Quizzes**

During weeks 3–13 (with the exception of week 10, which has no tutorial classes), each tutorial will also incorporate a separate student assessment in the format of quizzes. Each individual quiz has a weighting of 1%, therefore, tutorial quizzes for INFS1609/INFS2609 have an overall weighting of 10%. The content of each quiz will primarily reflect the student materials covered by students in the previous week. The aim of this assessment is to ensure that students are engaged in continual learning throughout the semester, and are prepared for each tutorial class.

**Programming Assignments**

There are three major assignments for this course. Each assignment will incorporate a small to medium size programming task. Assignment 1 has a weighting of 9%, assignment 2 has a weighting of 10%, and assignment 3 has a weighting of 11%. Therefore, programming assignments for INFS1609/INFS2609 have a weighting of 30% in total.

Through each assignment students will demonstrate their ability to understand and implement a range of technical skills relevant to the course. The assignments 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 in order to complete these assignments.

Assignment Submission Dates

Assignment 1: [Week 5] Wednesday 26 August, 5PM (Strict Deadline)
Assignment 2: [Week 9] Wednesday 23 September, 5PM (Strict Deadline)
Assignment 3: [Week 13] Wednesday 28 October, 5PM (Strict Deadline)

All three Assignments will be individual programming / assessments.

Further information regarding the specific details and submission procedure for assignments will be posted on Moodle as well as discussed during lectures and/or tutorials. However, before submitting assignments, students must ensure that their program compiles and runs as intended. All solutions for programming assignments will require appropriate use of commenting and naming conventions.

Please Note: Each student is responsible for their own work. Assignments will be checked for evidence of plagiarism. Plagiarism includes copying, inappropriate paraphrasing, collusion, as well as self-plagiarism. For instance, deliberately or recklessly presenting your work in collusion with others, copying or stealing another student’s assignment, or paying for work to be done may all be considered acts of Level 3 Plagiarism. This constitutes as significant plagiarism and serious student misconduct that is in breach of the Student Code.

Reminder: All students should keep a copy of all work submitted for assignments.

Final Examination

A final 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. Candidates may not bring any course materials to the examination. The examination paper may not be retained by the candidate.

4.3 Assignment Submission Procedure

Assignments need to be submitted online on the due day. A signed cover page must accompany submission of assignments. Digital signatures are not allowed. Missing cover page or cover page without authentic signatures will result in a penalty of 10% of the maximum marks available for assignments. Assignments will be screened with plagiarism-detecting software. The submission of non-original materials will be considered plagiarism and will be pursued.

4.4 Late Submission

It is your responsibility to adhere to the procedures for submission of assignments otherwise a penalty may apply. The key requirements are:

- Assignments shall be submitted as indicated in the course schedule and according to the instructions of the lecturer-in-charge.
- The late submission of assignments carries a penalty of 10% of the maximum marks for that assignment per day of lateness (including weekends and public holidays), unless an extension of time has been granted. For example, an
assignment worth 10% will attract a 1-mark penalty per day. An extension in the
time of submission will only be granted by the lecturer-in-charge for exceptional
circumstances, such as misadventure or illness. There are also provisions for
Special Consideration – see later in PART B “Special Consideration”.
Applications should be made to the lecturer-in-charge by email or in person.
You will be required to substantiate your application with appropriate
documentary evidence such as medical certificates, accident reports etc.
Please note that work commitments and computer failures are usually
considered insufficient grounds for an extension.
- **Partial submissions of your assignments will not be accepted.**

---

### 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 textbook for this course are:
*Object First with Java: A Practical Introduction Using BlueJ*
by David Barnes and Michael Kölling

---

## 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 Course and Teaching Evaluation and Improvement (CATEI) Process is one of
the ways in which student evaluative feedback is gathered. In this course, we will seek
your feedback through end of semester CATEI evaluations.
## COURSE SCHEDULE

<table>
<thead>
<tr>
<th>Week</th>
<th>Topic</th>
<th>Tutorial</th>
<th>Assessment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Week 1</td>
<td>Introduction &amp; Procedural Programming</td>
<td>No tutorial</td>
<td></td>
</tr>
<tr>
<td>27 July</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Week 2</td>
<td>Data Types &amp; Methods</td>
<td>Tutorial 1</td>
<td></td>
</tr>
<tr>
<td>3 August</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Week 3</td>
<td>Conditions and Loops &amp; Strings</td>
<td>Tutorial 2</td>
<td></td>
</tr>
<tr>
<td>10 August</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Week 4</td>
<td>Arrays</td>
<td>Tutorial 3</td>
<td></td>
</tr>
<tr>
<td>17 August</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Week 5</td>
<td>Sorting</td>
<td>Tutorial 4</td>
<td>Assignment 1 Submission Due</td>
</tr>
<tr>
<td>24 August</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Week 6</td>
<td>Modular Programming, Testing and Debugging</td>
<td>Tutorial 5</td>
<td></td>
</tr>
<tr>
<td>31 August</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Week 7</td>
<td>Multi-dimensional Arrays and Array Lists</td>
<td>Tutorial 6</td>
<td></td>
</tr>
<tr>
<td>7 September</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Week 8</td>
<td>Object Oriented Programming, Objects and Classes</td>
<td>Tutorial 7</td>
<td></td>
</tr>
<tr>
<td>14 September</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Week 9</td>
<td>Method Overloading and Serialization</td>
<td>Tutorial 8</td>
<td>Assignment 2 Submission Due</td>
</tr>
<tr>
<td>21 September</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Mid-semester break: Saturday 26 September – Monday 5 October inclusive</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Week 10</td>
<td>OO Design: Encapsulation and Exceptions (Monday 5 Oct is a public holiday)</td>
<td>No Tutorial</td>
<td></td>
</tr>
<tr>
<td>5 October</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Week 11</td>
<td>OO Design: Inheritance and Interfaces</td>
<td>Tutorial 9</td>
<td></td>
</tr>
<tr>
<td>12 October</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Week 12</td>
<td>OO Design: Polymorphism &amp; Revision</td>
<td>Tutorial 10</td>
<td></td>
</tr>
<tr>
<td>19 October</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Week 13</td>
<td>NO LECTURES</td>
<td>Tutorial 11</td>
<td>Assignment 3 Submission Due</td>
</tr>
<tr>
<td>26 October</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>