



# Course Outline (Higher Education)

<b>School:</b>	School of Engineering, Information Technology and Physical Sciences
<b>Course Title:</b>	AGILE CODING
<b>Course ID:</b>	ITECH2306
<b>Credit Points:</b>	15.00
<b>Prerequisite(s):</b>	ITECH1400 OR (ITECH2000 and ITECH2001)
<b>Co-requisite(s):</b>	Nil
<b>Exclusion(s):</b>	(ITECH2100)
<b>ASCED:</b>	020103

## Description of the Course:

This course introduces students to written programming code constructs used in the creation of a software system. Students must have demonstrated previous pre-requisite knowledge regarding logical problem solving and basic algorithmic development. Using an agile and requirement based approach to development, students will be introduced to the use of design modelling tools and an integrated development environment (IDE) to develop and edit source code, then compile, debug, test and run a software application. Students will be exposed to principles involved in engineering, designing and developing applications for information systems using object oriented and event driven paradigms.

**Grade Scheme:** Graded (HD, D, C, P, MF, F, XF)

## Work Experience:

No work experience: Student is not undertaking work experience in industry.

**Placement Component:** No

**Supplementary Assessment:** Yes

Where supplementary assessment is available a student must have failed overall in the course but gained a final mark of 45 per cent or above and submitted all major assessment tasks.

## Program Level:

Level of course in Program	AQF Level of Program					
	5	6	7	8	9	10
Introductory						

Level of course in Program	AQF Level of Program					
	5	6	7	8	9	10
Intermediate	■	■	✓	■	■	■
Advanced	■	■	■	■	■	■

### Learning Outcomes:

#### Knowledge:

- K1.** Explain agile software development principles.
- K2.** Describe written programming constructs used to create iteration, selection and modular code.
- K3.** Explain the object oriented principles of inheritance, polymorphism, encapsulation, dynamic binding and abstraction.
- K4.** Explain the principles of event-driven programming.

#### Skills:

- S1.** Develop object-oriented programs involving several interacting classes.
- S2.** Develop, compile and debug code using an appropriate IDE
- S3.** Manage multiple versions of source code used to create separate software releases.
- S4.** Select appropriate ways to represent data and collections of data.
- S5.** Incorporate pre-written classes, including those from an API, into software solutions
- S6.** Develop object oriented programs which incorporate event driven aspects.

#### Application of knowledge and skills:

- A1.** Design, develop, test, debug and release programs from supplied program specifications.

#### Course Content:

Using a blend of collaborative, problem based and online learning approaches, students will use industry standard Software development tools to plan, develop, implement and release software solutions.

Topics may include:

- Using an IDE to edit, develop, trace and debug code.
- Using object oriented modelling conceptualize an abstract design for an information system.
- Written programming constructs used for iteration and selection.
- Appropriate use of data types to store information within a program.
- Event driven software systems.
- Code design and modularization using object oriented approach.

- Source code control systems.
- Object-oriented programming concepts: encapsulation, polymorphism, inheritance, dynamic binding and abstraction.
- Design principles: classes and their relationships.

### Values:

- V1.** Appreciate the need for a professional and ethical approach to coding.
- V2.** Develop problem solving skills leading to self-efficacy.
- V3.** Appreciate the benefits of iterative development.

### Graduate Attributes

The Federation University FedUni graduate attributes (GA) are entrenched in the [Higher Education Graduate Attributes Policy](#) (LT1228). FedUni graduates develop these graduate attributes through their engagement in explicit learning and teaching and assessment tasks that are embedded in all FedUni programs. Graduate attribute attainment typically follows an incremental development process mapped through program progression. **One or more graduate attributes must be evident in the specified learning outcomes and assessment for each FedUni course, and all attributes must be directly assessed in each program**

Graduate attribute and descriptor		Development and acquisition of GAs in the course	
		Learning Outcomes (KSA)	Assessment task (AT#)
GA 1 Thinkers	Our graduates are curious, reflective and critical. Able to analyse the world in a way that generates valued insights, they are change makers seeking and creating new solutions.	S1, S4, S5, S6	AT1
GA 2 Innovators	Our graduates have ideas and are able to realise their dreams. They think and act creatively to achieve and inspire positive change.	S1, S2, S3, S4, S5, S6	AT1
GA 3 Citizens	Our graduates engage in socially and culturally appropriate ways to advance individual, community and global well-being. They are socially and environmentally aware, acting ethically, equitably and compassionately.	Not applicable	Not applicable
GA 4 Communicators	Our graduates create, exchange, impart and convey information, ideas, and concepts effectively. They are respectful, inclusive and empathetic towards their audience, and express thoughts, feelings and information in ways that help others to understand.	K2, S3	AT1, AT2
GA 5 Leaders	Our graduates display and promote positive behaviours, and aspire to make a difference. They act with integrity, are receptive to alternatives and foster sustainable and resilient practices.	Not applicable	Not applicable

### Learning Task and Assessment:

Students should engage with laboratory exercises and working with their peers, complete programming exercises to practise and develop their programming skills. Using an agile development methodology, students

will work individually and/or collaboratively to design, create (code) and test software. Students should engage with material made available online, in lectures or workshops and computer laboratory classes in order to support their knowledge acquisition and practical work. The assessment for the subject will include one practical assignment involving the development of an application. This assignment will be developed in stages. Developed solutions to smaller practical problems may be also assessed. The examination will test the understanding of the concepts and skills studied in this course.

Learning Outcomes Assessed	Learning Tasks	Assessment Type	Weighting
S1, S2, S3, S4, S5, S6, A1	Students will utilise object oriented programming constructs, pre-written-classes, algorithms, design constructs and standard style concepts to design, write, implement and document solutions to moderately simple to intermediate programming problems.	Projects / Assignments including at least one assignment done in a group	60-70%
K1, K2, K3, K4, S4, S5, A1	Students will provide theoretical answers and provide practical solutions to a range of questions and problem types drawn from theory and examples used during this course.	Tests / Examination(s)	30-40%

#### Adopted Reference Style:

APA

Refer to the [library website](#) for more information

Fed Cite - [referencing tool](#)