

Unit Outline (Higher Education)

Institute / School: Institute of Innovation, Science & Sustainability

Unit Title: AGILE CODING

Unit ID: ITECH2306

Credit Points: 15.00

Prerequisite(s): (ITECH2000 and ITECH2001) OR (ITECH1400)

Co-requisite(s): Nil

Exclusion(s): (ITECH2100)

ASCED: 020103

Description of the Unit:

This unit introduces students to programming constructs of a textual programming language used in the creation of a software system. Students must have demonstrated previous pre-requisite knowledge regarding logical problem solving and basic algorithmic development. Students will be introduced to the use of 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 the engineering/designing and developing of applications for information systems to address requirements using object-oriented and event-driven paradigms. Tools that enable collaborative coding are also presented.

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 Unit but gained a final mark of 45 per cent or above, has completed all major assessment tasks (including all sub-components where a task has multiple parts) as specified in the Unit Description and is not eligible for any other form of supplementary assessment

Course Level:

| Level of Unit in Course | AQF Level of Course | | | | | |
|-------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|
| | 5 | 6 | 7 | 8 | 9 | 10 |
| Introductory | <input type="checkbox"/> |
| Intermediate | <input type="checkbox"/> | <input type="checkbox"/> | ✓ | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Advanced | <input type="checkbox"/> |

Learning Outcomes:

Knowledge:

- K1.** Describe programming constructs used to create iteration, selection and modular code.
- K2.** Explain the object-oriented principles of inheritance, polymorphism, encapsulation, dynamic binding and abstraction.
- K3.** Explain the difference in how primitive and reference type variables are treated in memory
- K4.** Explain the principles of event-driven programming as it applies to the development of graphical user interfaces

Skills:

- S1.** Select and justify appropriate data types to represent information in a program.
- S2.** Develop object-oriented programs involving several interacting classes.
- S3.** Develop, compile, test and debug code using an appropriate Integrated Development Environment (IDE)
- S4.** Incorporate pre-written classes, including those from an API, into software solutions
- S5.** Manage code assets through use of a source-code version control system repository.
- S6.** Develop object-oriented programs which incorporate event driven aspects.
- S7.** Analyse and explain the behaviour of code which someone else has written

Application of knowledge and skills:

- A1.** Design, develop, test and debug programs from supplied program specifications.
- A2.** Use source-code version control tools to manage a repository of code

Unit Content:

Students will learn how to code in a current object-oriented programming language. They will learn how to use modern industry-standard software development tools to plan, develop/implement, test and manage software solutions.

Topics may include:

- Written programming constructs used for iteration and selection.
- Appropriate use of data types to store information within a program.
- Object-oriented programming concepts: encapsulation, polymorphism, inheritance, dynamic binding and abstraction.
- Code design and modularization using object-oriented approach.
- Design principles: classes and their relationships.
- Using object-oriented modelling conceptualize an abstract design for an information system.
- Using an IDE to edit, develop, trace and debug code.
- Event-driven Graphical User Interfaces

- Source code version-control systems.

FEDTASKS

Federation University Federation recognises that students require key transferable employability skills to prepare them for their future workplace and society. FEDTASKS (**T**ransferable **A**tttributes **S**kills and **K**nowledge) provide a targeted focus on five key transferable Attributes, Skills, and Knowledge that are embedded within curriculum, developed gradually towards successful measures and interlinked with cross-discipline and Co-operative Learning opportunities. *One or more FEDTASK, transferable Attributes, Skills or Knowledge must be evident in the specified learning outcomes and assessment for each FedUni Unit, and all must be directly assessed in each Course.*

| FEDTASK attribute and descriptor | | Development and acquisition of FEDTASKS in the Unit | |
|---|---|---|-----------------------|
| | | Learning Outcomes (KSA) | Assessment task (AT#) |
| FEDTASK 1 Interpersonal | Students will demonstrate the ability to effectively communicate, interact and work with others both individually and in groups. Students will be required to display skills in-person and/or online in: <ul style="list-style-type: none"> • Using effective verbal and non-verbal communication • Listening for meaning and influencing via active listening • Showing empathy for others • Negotiating and demonstrating conflict resolution skills • Working respectfully in cross-cultural and diverse teams. | Not applicable | Not applicable |
| FEDTASK 2 Leadership | Students will demonstrate the ability to apply professional skills and behaviours in leading others. Students will be required to display skills in: <ul style="list-style-type: none"> • Creating a collegial environment • Showing self-awareness and the ability to self-reflect • Inspiring and convincing others • Making informed decisions • Displaying initiative | Not applicable | Not applicable |
| FEDTASK 3 Critical Thinking and Creativity | Students will demonstrate an ability to work in complexity and ambiguity using the imagination to create new ideas. Students will be required to display skills in: <ul style="list-style-type: none"> • Reflecting critically • Evaluating ideas, concepts and information • Considering alternative perspectives to refine ideas • Challenging conventional thinking to clarify concepts • Forming creative solutions in problem solving | S2, S3, S6, A1 | AT2, AT3, AT4 |
| FEDTASK 4 Digital Literacy | Students will demonstrate the ability to work fluently across a range of tools, platforms and applications to achieve a range of tasks. Students will be required to display skills in: <ul style="list-style-type: none"> • Finding, evaluating, managing, curating, organising and sharing digital information • Collating, managing, accessing and using digital data securely • Receiving and responding to messages in a range of digital media • Contributing actively to digital teams and working groups • Participating in and benefiting from digital learning opportunities | S3, S5, A2 | AT2, AT3, AT4 |

| FEDTASK attribute and descriptor | | Development and acquisition of FEDTASKS in the Unit | |
|--|---|---|-----------------------|
| | | Learning Outcomes (KSA) | Assessment task (AT#) |
| FEDTASK 5 Sustainable and Ethical Mindset | Students will demonstrate the ability to consider and assess the consequences and impact of ideas and actions in enacting ethical and sustainable decisions. Students will be required to display skills in: <ul style="list-style-type: none"> • Making informed judgments that consider the impact of devising solutions in global economic environmental and societal contexts • Committing to social responsibility as a professional and a citizen • Evaluating ethical, socially responsible and/or sustainable challenges and generating and articulating responses • Embracing lifelong, life-wide and life-deep learning to be open to diverse others • Implementing required actions to foster sustainability in their professional and personal life. | Not applicable | Not applicable |

Learning Task and Assessment:

Students should engage with the weekly laboratory exercises to develop their programming skills. In class, students will work individually to plan, create (code) and test software to gain practice in programming. Students should engage with all material made available online, in lectures and computer laboratory classes in order to support their knowledge acquisition and practical work.

The in-semester assessment for the subject will involve the development and testing of working programs, drawing together topics from multiple weeks. The final test/examination will test the understanding of the concepts and skills studied in this course.

| Learning Outcomes Assessed | Assessment Tasks | Assessment Type | Weighting |
|--|---|--|-----------|
| K1, K2, K3, K4, S1, S6, S7, A1 | Provide theoretical answers and provide practical solutions to a range of questions and problem types drawn from theory and topics used during this unit, in one or more quizzes. | Quiz or Quizzes | 0-20% |
| S1, S2, S3, S4, S5, S6, S7, A1 | Students will be required to complete tasks during weekly lab sessions. Many of these labs will require the student to either demonstrate how the code they have written fulfils an objective of the task, or to explain what portions of the code achieve or how a particular concept has been implemented, or to make an on-the-spot alteration requested by the lab tutor. | Lab work demonstrations and explanations | 0-30% |
| S1, S2, S3, S4, S5, S6, S7, A1, A2, K1, K2, K4 | Students will develop moderate sized multi-class programs possibly requiring association and/or inheritance relationships between classes, and using other techniques taught (including possibly the integration of file IO or GUI aspects), to fulfill a set of provided requirements. They may also be asked to create a video demonstrating that they can use git and other taught techniques. They may be required to use a git repository which may provide 'starting' code. It is anticipated that at least 2 major programming assignments will be issued. | Programming Assignments | 50-80% |

Adopted Reference Style:

APA

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

Fed Cite - [referencing tool](#)