# Software Engineering 11–12 (2022): Sample scope and sequence (Year 11) (120 hours)

## Term 1

| Weeks 1–8\* | Weeks 9–10\* |
| --- | --- |
| **Unit 1:** Beginning Programming  **Focus area(s):** Programming Fundamentals  This unit introduces students to computer programming. It explores base concepts of understanding requirements, development and testing stages from the perspective of both a user and a developer.  Students also explore the development and use of algorithms, data, data types, data structures and the role design plays in the development of code and the final program.  An understanding and application of the knowledge and techniques learnt in this unit will be applied to the development and construction of solutions to a variety of different problems.  Students apply backup and maintenance procedures to activities that are computer based. Activities include the development and application of algorithms, appropriate programming code, suitable user and developer documentation, and an effective user interface (UI).  Small practical activities using the Python programming language to develop skills in converting algorithms and computational thinking into functional applications.  The use of a suitable **Python IDE** will be required (such as Wing–Free Edu Version available, Spyder or PyCharm) to create, document and test the developed programming solution(s). | **Unit 2:** Object-Oriented Programming  **Focus area(s):** The Object-Oriented Paradigm  This unit extends the understanding of basic programming structures into the Object-Oriented Paradigm (OOP). Key concepts such as objects, classes, inheritance and polymorphism will be explored.  The application of these key concepts and others will be used to create, test and modify OOP-based software solutions. The use of version control and backup systems should also be employed for all projects.  The activities undertaken in this unit will build upon the concepts from Unit 1 and develop the concepts of OOP-based design and coding in the creation of more complex solutions.  The use of a suitable **Python IDE** will be required (such as Wing–Free Edu Version available, Spyder or PyCharm) to create, document and test the developed programming code and applications.  A journal or logbook may be introduced as a method of recording progress throughout the development process.  The journal or logbook could be digital or non-digital and enable students to record their ideas in writing, using images, audio or multimodally. |
| Outcomes: **SE-11-01, SE-11-02, SE-11-03, SE-11-04, SE-11-06, SE-11-07** | Outcomes: **SE-11-01, SE-11-02, SE-11-03, SE-11-04, SE-11-06, SE-11-07, SE-11-08, SE-11-09** |

## Term 2

|  |  |
| --- | --- |
| Weeks 1–8\* | Weeks 9–10\* |
| **Unit:** Object-Oriented Programming (continued)  **Focus area(s):** The Object-Oriented Paradigm  This unit extends the understanding of basic programming structures into the Object-Oriented Paradigm (OOP). Key concepts such as objects, classes, inheritance and polymorphism will be explored. The application of these key concepts and others will be used to create, test and modify OOP-based software solutions. The use of version control and backup systems should also be employed for all projects.  The activities undertaken in this unit will build upon the concepts from Unit 1 and develop the concepts of OOP-based design and coding in the creation of more complex solutions.  The use of a suitable **Python IDE** will be required (such as Wing–Free Edu Version available, Spyder or PyCharm) to create, document and test the developed programming code and applications. A journal or logbook may be introduced as a method of recording progress throughout the development process. The journal or logbook could be digital or non-digital and enable students to record their ideas in writing, using images, audio or multimodally. | **Unit:** Mechatronics  **Focus area(s):** Programming Mechatronics  This unit of work will focus on the application of procedural programming and OOP in a robotic and mechatronic environment and will focus on the direct relationship between the hardware items associated with a mechatronic system (sensor, effectors and actuators) and the software that is required to control the interactions.  The use of a suitable **Python IDE** will be required (such as Wing–Free Edu Version available, Spyder or PyCharm) to create, document and test the developed programming code and applications. Students are encouraged to use a journal or logbook as a method of recording progress throughout the development process. The journal or logbook could be digital or non-digital and enable students to record their ideas in writing, using images, audio or multimodally. |
| Outcomes: **SE-11-01, SE-11-02, SE-11-03, SE-11-04, SE-11-06, SE-11-07, SE-11-08, SE-11-09** | Outcomes: **SE-11-01, SE-11-02, SE-11-03, SE-11-04, SE-11-05, SE-11-06, SE-11-07, SE-11-08, SE-11-09** |

|  |
| --- |
| Weeks 1–8\* |
| **Unit:** Mechatronics (continued)  **Focus area(s):** Programming Mechatronics  This unit of work will focus on the application of procedural programming and OOP in a robotic and mechatronic environment and will concentrate on the relationship between the hardware items associated with a mechatronic system (sensor, effectors and actuators) and the software that is required to control the interactions.  The use of a suitable **Python IDE** will be required (such as Wing–Free Edu Version available, Spyder or PyCharm) to create, document and test the developed programming code and applications. Students are encouraged to use a journal or logbook as a method of recording progress throughout the development process. The journal or logbook could be digital or non-digital and enable students to record their ideas in writing, using images, audio or multimodally. |
| Outcomes: **SE-11-01, SE-11-02, SE-11-03, SE-11-04, SE-11-05, SE-11-06, SE-11-07, SE-11-08, SE-11-09** |

## Term 3

\* *Week calculations are based on standard 10-week terms. Term 3 has considered the option for a 2-week assessment/examination block in a standard   
10-week term calendar at the end of Year 11.*