Unit 1 Applied Computing 2025

Outcome 2 Programming – Template for developing an assessment task – Plan

Outcome 2 On completion of this unit the student should be able to interpret teacher-provided solution requirements to design and develop a software solution using an object-oriented programming language.		**Assessment task development** Create a scenario that is a real-world example that provides students with solution requirements for them to design and develop using a range of appropriate features of an OOP language and then test and debug the software solution. Key content within the assessment task should be based on the targeted key knowledge and key skills.
Key knowledge	**Key skills**	
• emerging trends in programming languages and artificial intelligence-based (AI) code generators for the development of software solutions, such as: – low-code development approaches – readability and/or maintainability improvements • characteristics of functional and non-functional requirements, constraints and scope • key legal requirements relating to intellectual property and copyright while designing and developing software	• interpret solution requirements to develop a software solution	Content to be included in the assessment task should introduce students to a scenario. The scenario should provide students with solution requirements (functional and non-functional) to develop the software solution. The scenario should enable students to demonstrate their knowledge and to meet the requirements of the outcome.
• design tools for representing the functionality and appearance of solution designs, such as: – mock-ups – input-process-output (IPO) charts – flowcharts/pseudocode	• select and use appropriate design tools to represent solution designs	The scenario with the solution requirements should enable students to determine how they will select and use a range of appropriate design tools. Designs are to represent the appearance of their software solution. Teachers are not to provide the designs for students.
• characteristics of data types, such as: – text (character, string) – numeric (integer, floating point, date/time) – Boolean • types of data structures, such as: – one-dimensional arrays – lists – records (varying data types, field index)	• use a range of data types and data structures	The scenario with the solution requirements should enable students to determine what data types and data structures they will need to use for the software solution.
• principles of OOP, such as: – abstraction – encapsulation • features of a programming language, such as: – variables, and initialising, accessing and storing data in variables – control structures (sequence, selection and iteration/repetition) – arithmetic, logical and conditional operators – procedures, functions and methods	• develop a software solution using appropriate features of an OOP language	The scenario with the solution requirements should enable students to determine the appropriate features of an OOP language and use of naming conventions they will need to develop the software solution. An appropriate OOP language should be used by the students.

Unit 1 Applied Computing 2025 Outcome 2 Programming – Template for developing an assessment task – Plan		
• naming conventions for solution elements, such as: – Hungarian notation – camel casing		
• purposes of internal documentation, such as: – explaining data and code structures – code maintenance	• document the functioning of a software solution through internal documentation	Students are to write internal documentation within the OOP language to document the functioning of the software solution.
• validation techniques for data, such as: – existence checking – type checking – range checking • debugging and testing techniques for checking software solutions function correctly, such as: – test tables to compare expected and actual output – construction of relevant test data – breakpoints – debugging output statements	• design and apply suitable validation, debugging and testing techniques	Students are to apply suitable validation techniques when developing the software solution. They are to design a testing table and use suitable testing techniques to determine the expected results of testing. The testing table should include test data, objects and processing such as calculations, etc. The testing table should also include a column for the actual results of testing. Suitable debugging techniques should be applied to ensure all the tests of the software solution meet the solution requirements.