

Unit 1 Applied Computing – 2024
Outcome 2 Programming – Template for developing an assessment task – Plan

| Outcome 2 | | Assessment task development |
|---|---|--|
| On completion of this unit the student should be able to interpret teacher-provided solution requirements to design, develop and evaluate a software solution using a programming language. | | Create a scenario that is a real-world example that provides students with solution requirements, constraints and scope for them to design and develop using a range of processing features of a programming language and then test and debug the software solution. Students are to document and monitor a project plan for the duration of the assessment task. Key content within the assessment task should be based on the targeted key knowledge and key skills. |
| Key knowledge | Key skills | |
| <ul style="list-style-type: none"> functions and capabilities of key hardware and software components of digital systems required for processing, storing and communicating data and information features of functional and non-functional solution requirements, constraints and scope | <ul style="list-style-type: none"> analyse 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), constraints and scope to develop the software solution. The scenario should enable students to demonstrate their knowledge and to meet the requirements of the outcome. |
| <ul style="list-style-type: none"> design tools for representing the functionality and appearance of solution designs such as data dictionaries, mock-ups and pseudocode | <ul style="list-style-type: none"> 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 (data dictionaries, mock-ups and pseudocode). Designs are to represent the appearance of their software solution. Teachers are not to provide the designs for students. |
| <ul style="list-style-type: none"> characteristics of data types types of data structures formatting and structural characteristics of input and output such as file formats | <ul style="list-style-type: none"> 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. |
| <ul style="list-style-type: none"> naming conventions for solution elements such as files, functions, methods and variables processing features of a programming language characteristics of internal documentation | <ul style="list-style-type: none"> develop a software solution using appropriate processing features of a programming language | The scenario with the solution requirements should enable students to determine the appropriate processing features, naming conventions and use of internal documentation they will need to develop the software solution. An appropriate programming language should be used by the students. |
| <ul style="list-style-type: none"> testing and debugging techniques to ensure software solutions meet requirements such as test tables and test data | <ul style="list-style-type: none"> design and apply suitable testing and debugging techniques using appropriate test data | Students 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 (if used) 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. |
| <ul style="list-style-type: none"> techniques for evaluating the efficiency and effectiveness of software solutions | <ul style="list-style-type: none"> evaluate the efficiency and effectiveness of the software solution to meet requirements | Students are to evaluate the efficiency and effectiveness of their software solution in meeting the teacher-provided solution requirements. Definitions for efficiency and effectiveness can be found in the <i>Terms used in this study</i> section of the <i>Applied Computing Study Design</i> . This evaluation could be completed as a written report. |
| <ul style="list-style-type: none"> project plans to coordinate and monitor the tasks, including sequencing and time allocation to create software solutions | <ul style="list-style-type: none"> document and monitor project plans using software | Students are to document their project plans before commencing the project. Project plans are to include the sequencing and time allocation of the tasks required to develop the software solution. They are to record and monitor the progress of the project plan in the development of their software solution during the life of the project. |