**Unit 3 Software Development – 2024**

**Outcome 1 Software development: programming – Template for developing an assessment task – Blank**

| **Outcome 1**<br>On completion of this unit the student should be able to interpret teacher-provided solution requirements and designs, and apply a range of functions and techniques using a programming language to develop and test working software modules. | | | **Assessment task development** |
|---|---|---|---|
| **Key knowledge** | **Key skills** | **VCAA Performance descriptors (Very high)** | |
| • methods for documenting a problem, need or opportunity<br>• methods for determining solution requirements, constraints and scope<br>• methods of representing designs, including data dictionaries, mock-ups, object descriptions and pseudocode | • interpret solution requirements and designs to develop working modules | • All solution requirements and designs are interpreted accurately to developing working modules. | |
| • characteristics of data types<br>• types of data structures, including associative arrays (or dictionaries or hash tables), one-dimensional arrays (single data type, integer index) and records (varying data types, field index)<br>• formatting and structural characteristics of files, including delimited (CSV), plain text (TXT) and XML file formats | • use a range of data types and data structures | • Comprehensive selection of relevant data types and data structures to develop working modules. | |
| • a programming language as a method for developing working modules that meet specified needs<br>• naming conventions for solution elements<br>• processing features of a programming language, including classes, control structures, functions, instructions and methods<br>• algorithms for sorting, including selection sort and quick sort<br>• algorithms for binary and linear searching | • use and justify appropriate processing features of a programming language to develop working modules | • Comprehensive selection and use of relevant processing features of the programming language to develop all working modules.<br>• Comprehensive justification and explanation of how the selection of appropriate processing features of the programming language are used to develop working modules. | |
| • validation techniques, including existence checking, range checking and type checking<br>• techniques for checking that modules meet design specifications, including trace tables and construction of test data | • develop and apply suitable validation, testing and debugging techniques using appropriate test data | • Comprehensive use of relevant data validation techniques are applied efficiently and effectively to check the reasonableness of all input data.<br>• Comprehensive use of test data is expressed in a testing table, with both expected and actual output stated, and showing detailed evidence of debugging. | |
| • purposes and characteristics of internal documentation, including meaningful comments and syntax | • document the functioning of modules and the use of processing features through internal documentation | • All software modules include comprehensive internal documentation regarding the functioning of modules and use of selected processing features. | |