Unit 3 Software Development – 2024

Outcome 1 Software development: programming – Developing a marking scheme – Sample

Outcome 1 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.			**Developing a marking scheme – Marks allocated – 100** Refer to the key skills or the VCAA performance descriptors when developing a marking scheme for the assessment task. Determine the weighting of the marks out of 100 for each key skill or performance descriptor. When determining weightings consider the time that students will take to complete each task as well as the level of difficulty of each task. Marks should be allocated to ensure students can demonstrate a range of levels of performance in the task.
Key knowledge	**Key skills**	**VCAA Performance descriptors (Very high)**	
• methods for documenting a problem, need or opportunity • methods for determining solution requirements, constraints and scope • 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.	Students are to interpret the solution requirements and designs for between three and six working modules. Possible number of marks – 10 marks
• characteristics of data types • 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) • 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.	Students are to use a range of relevant data types and data structures within their software modules. Possible number of marks – 10 marks
• a programming language as a method for developing working modules that meet specified needs • naming conventions for solution elements • processing features of a programming language, including classes, control structures, functions, instructions and methods • algorithms for sorting, including selection sort and quick sort • 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. • Comprehensive justification and explanation of how the selection of appropriate processing features of the programming language are used to develop working modules.	Students are to use appropriate processing features, naming conventions and sorting and searching algorithms to develop their software modules. A higher weighting of marks should be included to meet this key skill or performance descriptor. Possible number of marks – 40 marks Students are to justify and explain their selection of processing features and sorting and searching algorithms used to develop their working modules. Possible number of marks – 10 marks
• validation techniques, including existence checking, range checking and type checking • 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. • Comprehensive use of test data is expressed in a testing table, with both expected and actual output stated, and showing detailed evidence of debugging.	Students are to use and apply relevant data validation techniques to check all input data. Possible number of marks – 10 marks Students test their working modules using appropriate testing techniques. Possible number of marks – 10 marks
• 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.	Students are to include internal documentation within their working modules. Possible number of marks – 10 marks