Title	**Apply knowledge and concepts of software development to create applications**		
Level	**4**	**Credits**	**15**

Purpose	People credited with this unit standard are able to: demonstrate and apply knowledge and concepts of software development tools and techniques; apply mathematical and logical techniques and problem-solving skills to solve computational problems; and apply principles of software development and relevant project management methodologies to create and test interactive working applications, integrating mathematical and logical techniques. This unit standard has been developed primarily for assessment within programmes leading to the New Zealand Certificate in Information Technology Essentials (Level 4) [Ref: 2594]. It is intended for people who want to develop the essential skills for further study that will equip them to work in the field of Information Technology (IT) as a profession, and who will generally not have previous computer programming experience such as digital technologies achievement standards.

Classification	Computing > Software Development - Programming

Available grade	Achieved

Guidance Information

1 Recommended skills and knowledge:
Unit 18740, *Create a simple computer program to meet a set brief*, or Unit 18741 *Create a computer program to provide a solution to a problem,* and Unit 29778, *Use the main features and functions of a schematic diagram application to create diagrams*, or demonstrate equivalent knowledge and skills.

2 Assessment, where applicable, will be conducted in and for the context of real or realistic situations and/or settings, and be relevant to current and/or emerging practice. The assessor may gather evidence over time from a range of scenarios rather than using one assessment where the learner has to demonstrate all of the required skills.

The assessment context for this unit standard must be suitable to meet the criteria for Level 4 in the NZQF Level Descriptors, which are available by searching for “level descriptors” at www.nzqa.govt.nz.

3 A brief will be supplied to the learner. The brief will outline the problem that the program is intended to solve, and will include a clear description of the desirable outcomes sought, the constraints to be met by the solution, and specifications and requirements against which the success or otherwise of the program can be evaluated. The specifications will include operating and testing procedures, and requirements for input, processing, and output.

The test plan must cover both valid, boundary and invalid inputs, and will describe various test scenarios, and list the expected outcomes. The program specification will define what range of data is valid, and the action(s) that should be taken when invalid data is input.

4 Performance of all outcomes is to be carried out in accordance with good practice. In software development, good practice refers to conventions used and recommended by an organisation involved in the software development, computing or IT industry in New Zealand. It includes coding standards such as using meaningful variable names, indenting code to reflect program logic, and inserting comments to explain program logic

5 Definitions

An *algorithm* is a finite list of instructions or steps needed to solve a problem.
A *design plan* describes the module composition and logic/structure of the program being developed, and outlines how the requirements of the specifications will be realised.
Desk-checked is a manual process where a programmer visually inspects code (usually off-line with hard copy) checking for syntax and logic errors. This may require devising some basic input data in order to predict the program behaviour. Errors are located and corrected without help of compiler diagnostics or computer generated output.
Flowcharts are a diagrammatic representation of an algorithm or program. A flowchart can be helpful for both writing programs and explaining the program to others.
Organisation refers to a specific entity which may be – in private, public, or community and voluntary sectors; a business, a discretely managed unit within a larger entity, a Māori organisation, or other special-purpose body.
A *program* with a user-interface is an application, but many programs are not applications. An *application* program (or app for short) is any software program designed for an end user.
Project management methodologies are strictly defined combination of logically related practices, methods and processes that determine how best to plan, develop, control and deliver a project through to completion and termination. It is a systematic and disciplined approach to project design, execution and completion.
Pseudocode is a detailed yet readable description of what a computer program or algorithm must do, expressed in a formally styled natural language rather than in a programming language.

6 Legislation relevant to this unit standard includes but is not limited to the:
Copyright Act 1994
Copyright (New Technologies) Amendment Act 2008
Harmful Digital Communications Act 2015
Health and Safety at Work Act 2015
Privacy Act 2020
Unsolicited Electronic Messages Act 2007
and any subsequent amendments.
Current legislation and regulations can be accessed at http://legislation.govt.nz.

7 References
ACC5637 Guidelines for Using Computers - Preventing and managing discomfort, pain and injury. Accident Compensation Corporation - Department of Labour, 2010; available from Worksafe New Zealand, at https://www.worksafe.govt.nz/topic-and-industry/work-related-health/musculoskeletal-disorders/ergonomics/safely-using-computers-at-work/.
Parsons, D and Haden, P; *Parson's programming puzzles,* Otago Polytechnic, Dunedin; Proceeding ACE '06 Proceedings of the 8th Australasian Conference on Computing Education - Volume 52, Pages 157-163 available at https://dl.acm.org/doi/abs/10.5555/1151869.1151890.

Outcomes and performance criteria

Outcome 1

Demonstrate and apply knowledge and concepts of software development tools and techniques.

Performance criteria

1.1 Software development tools are identified, described, and compared in terms of their usage and application.

Range includes logic diagrams and pseudocode; programming languages.

1.2 Small blocks/sections of provided code are desk-checked to locate syntax and logic errors.

Range up to 20 lines of provided procedural code with a stated purpose;
provided code includes at least 5 syntax errors and 5 logic errors;
constructs must include sequence, iteration, and decision (if-then-else) elements;
at least 80% of the errors must be located and corrected.

1.3 A logic diagram is translated into logically correct and syntax-error-free code.

Range at least 10 elements in a provided logic diagram which include a loop and one or more decision elements;
includes code commenting;
may include the use of a compiler/interpreter to assist with syntax and logic error detection.

Outcome 2

Apply mathematical and logical techniques and problem-solving skills to solve computational problems.

Performance criteria

2.1 Mathematical and logical techniques are applied to solve supplied computational problems.

Range techniques must include arithmetic operators, logical operators, comparison operators, true/false, decimals, fractions, if statements, geometrical calculations;
evidence of at least five computational problems.

2.2 Procedural applications using simple algorithms are created to solve supplied programming problems.

Range algorithms may include but are not limited to – exchanging the values stored in two variables; re-ordering the values stored in three variables into ascending or descending order; finding the largest or smallest element in a collection (list, array, dictionary); finding the mean value of a set of numbers inputted by a user; determining whether a list of numbers is in ascending/descending sequence; determining if a year, input by the user, is a leap year; five different algorithms are required, with at least one employing an indexed data structure;
evidence of five programming problems.

Outcome 3

Apply principles of software development and relevant project management methodologies to create and test interactive working applications, integrating mathematical and logical techniques.

Range constructs must include – sequence; iteration; constants; variables; selection statements; input/output commands; collections (list, array, dictionary); data structures (indexed, stacks, queues); types; user defined functions/methods/procedures; user interface.
project management includes – iterative development; use of digital tools to track progress; version control.

Performance criteria

3.1 Programming concepts, techniques and tools are applied to design interactive procedural applications with a stated purpose using a suitable programming language and industry relevant project management methodologies.

Range may include but is not limited to – flowcharts, logic diagrams or pseudocode;
includes – decomposition of the problem; logical, mathematical, and decision-making techniques; a plan for testing the code with expected outcomes (valid, invalid and boundary cases).

3.2 Programming concepts and tools are applied to code interactive procedural applications with a stated purpose using a suitable language.

Range must include all standard operators (add, subtract, divide, multiply, exponentiation); logical operators (AND, OR, NOT) and brackets "()"; correct use of operator precedence; code commenting; following common conventions for the chosen programming language.

3.3 Code is tested following planned test plan, and the test outcomes are documented.

Range includes – valid, invalid, boundary, comparison of expected outcomes and actual outcomes.

3.4 The code is debugged and re-tested to eliminate errors and meet design plan and specifications.

Range all modifications are documented, including reasons for them.

Replacement information	This unit standard replaced unit standard 29813.

Planned review date	31 December 2026

Status information and last date for assessment for superseded versions

Process	Version	Date	Last Date for Assessment
Registration	1	28 April 2022	N/A

Consent and Moderation Requirements (CMR) reference	0099

This CMR can be accessed at http://www.nzqa.govt.nz/framework/search/index.do.

Comments on this unit standard

Please contact Toi Mai Workforce Development Council qualifications@toimai.nz if you wish to suggest changes to the content of this unit standard.