

Title	Demonstrate an understanding of the principles of computer programming		
Level	5	Credits	7

Purpose	<p>People credited with this unit standard are able to: describe computer program development life cycles; functionally decompose problems for computer programs; verify logical depictions for computer programs; compare the characteristics of common computer programming languages; demonstrate an understanding of the principles of object-oriented computer programming; and demonstrate an understanding of the principles of event-driven computer programming. The performance of all outcomes is to a standard that allows for further learning in this area.</p>
----------------	---

Classification	Computing > Software Development - Programming
-----------------------	--

Available grade	Achieved
------------------------	----------

Guidance Information

- 1 The term *industry recommended format* refers to a format used and recommended by an organisation involved in the computer industry.
- 2 The assessment context for this unit standard must be suitable to meet the criteria for level 5 in the NZQF Level Descriptors, which are available by searching for “level descriptors” at www.nzqa.govt.nz.
- 3 Legislation relevant to this unit standard may include 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 1993
 - Unsolicited Electronic Messages Act 2007
 and any subsequent amendments.
 Current legislation and regulations can be accessed at <http://legislation.govt.nz>.
- 4 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 <http://www.business.govt.nz/worksafe/information-guidance/all-guidance-items/guidelines-for-using-computers>.

Outcomes and performance criteria

Outcome 1

Describe computer program development life cycles.

Range current industry recommended development life cycles.

Performance criteria

- 1.1 The description distinguishes each stage in the life cycles.
- 1.2 The description outlines procedures for each stage in the life cycles.
- 1.3 The description outlines issues for each stage in the life cycles.

Outcome 2

Functionally decompose problems for computer programs.

Range decomposition must include at least four lowest level functions.

Performance criteria

- 2.1 Problem elements are depicted in a logically ordered form using an industry recommended format.
- 2.2 The decomposition is modularised, i.e. decomposed into sub-processes.
- 2.3 The decomposition exhibits functional granularity at the lowest level, i.e. cannot be further decomposed.

Outcome 3

Verify logical depictions for computer programs.

Range at least two industry recommended depiction methods.

Performance criteria

- 3.1 Boundary conditions are identified.
- 3.2 Errors in the logical depiction are identified and corrected.
- 3.3 The output of the logical depiction is derived.
- 3.4 The depiction is translated into an alternative form of depiction.

Outcome 4

Compare the characteristics of common computer programming languages.

Range at least three languages currently in use in commercial or government organisations.

Performance criteria

- 4.1 The comparison distinguishes languages and identifies examples of their application.
- 4.2 Decisions on whether to use a language for given situations are justified by describing the advantages and disadvantages of the language compared with the alternative languages.

Outcome 5

Demonstrate an understanding of the principles of object-oriented computer programming.

Performance criteria

- 5.1 An examination of a depiction of an object-oriented program identifies the components of the program.
- Range objects, methods, classes, class hierarchies.
- 5.2 A comparison of object-oriented programming with programming using procedural languages identifies and explains the advantages of object-oriented programming.

Outcome 6

Demonstrate an understanding of the principles of event-driven computer programming.

Performance criteria

- 6.1 An analysis of a description of computer programming requirements identifies the program events.
- 6.2 An examination of a depiction of an event-driven program identifies the components of the program.
- Range event messages, message handlers.
- 6.3 Decisions on whether to use event-driven programming for given situations are justified by describing the advantages and disadvantages of event-driven programming.

This unit standard is expiring. Assessment against the standard must take place by the last date for assessment set out below.

Status information and last date for assessment for superseded versions

Process	Version	Date	Last Date for Assessment
Registration	1	8 July 1996	31 December 2013
Revision	2	28 November 2000	31 December 2015
Rollover and Revision	3	19 September 2013	31 December 2019
Review	4	19 January 2017	31 December 2024
Review	5	26 May 2022	31 December 2024

Consent and Moderation Requirements (CMR) reference

0113

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