

<b>Title</b>	<b>Create a computer program to provide a solution</b>		
<b>Level</b>	<b>3</b>	<b>Credits</b>	<b>6</b>

<b>Purpose</b>	People credited with this unit standard are able to: formulate a design specification, write code, and test and modify a program to provide a solution to a problem; and create simple end-user documentation.
----------------	--

<b>Classification</b>	Computing > Software Development - Programming
-----------------------	--

<b>Available grade</b>	Achieved
------------------------	----------

<b>Entry information</b>	
<b>Recommended skills and knowledge</b>	Unit 18740, <i>Create a simple computer program to meet a set brief</i> , or demonstrate equivalent knowledge and skills.

### Explanatory notes

- 1 Candidates are required to create a structured computer program with a user interface that enables efficient program operation. Where possible, for moderation purposes, the program should be saved as an executable file. Where this is not possible an assessor attestation clearly indicating how the assessment judgement was made, along with any supporting evidence, must be supplied.
- 2 The program must include but is not limited to – keyboard input, screen output, looping, iteration, decisions/selection, arithmetic and data validation. Data validation requires the program to perform some testing on input data to ensure its validity. This could include but is not limited to – range-checking, numeric format, upper or lower case, input length, or other appropriate method.
- 3 A *plan* outlines how the requirements of the program will be realised. The plan must include the specifications and/or features required by the program to provide the solution. The plan may be modified during the task and changes justified. Evidence of planning may be oral, written, and/or graphic. Depending on the assessment context, the plan will include key milestone outcomes, and may include but is not limited to:
  - how resources such as time, expertise and materials (and finance, if appropriate) will be used to achieve the outcomes of each milestone;
  - how consultation with stakeholders will be carried out to ensure that all constraints and requirements are met.

#### 4 Definitions

*Internal documentation* means documentation included as comments within the source code, rather than documentation created separately.

*Problem decomposition* means breaking the problem down into smaller manageable components.

A *simple end-user document* is intended for a non-skilled user of a program, rather than providing documentation for a programmer. The end-user document must use consistent font and layout, be legible, and should either avoid the use of undefined jargon or acronyms, or provide a glossary for these. The document must be saved in a format that is accessible to users.

#### 5 Legislation relevant to this unit standard includes but is not limited to the:

Copyright Act 1994,

Copyright (New Technologies) Amendment Act 2008;

Health and Safety in Employment Act 1992,

and any subsequent amendments.

#### 6 An assessment resource to support computing unit standards (levels 1 to 4) can be found on the NZQA website at [www.nzqa.govt.nz/asm](http://www.nzqa.govt.nz/asm).

'*The Computing Process - a clarification document*' contains further information and can be found on the NZQA website.

---

## Outcomes and evidence requirements

### Outcome 1

Formulate a design specification for a computer program to provide a solution to a problem.

### Evidence requirements

1.1 A design specification is formulated that identifies the problem to be solved and describes the program in terms of purpose and target users.

1.2 The design specification includes a problem decomposition diagram that expresses the logical structure of the program and is tested for conformity to design requirements.

1.3 The design specification outlines the specifications (including constraints) to be met by the program in order to provide a solution.

Range specifications may include but are not limited to – computer language to be used, input and output requirements, outline of the proposed testing procedures.

1.4 A plan is developed to realise the design specification.

Range milestones, resources, stakeholder consultations and endorsement.

**Outcome 2**

Write code for the program.

**Evidence requirements**

- 2.1 Code is created in accordance with design specification and the program decomposition diagram.
- 2.2 Code is internally documented to meet the requirements of the design specifications.

Range meaningful descriptive names for variables, procedures, functions, labels and data files; use of indentation and line spacing; use of comments to explain program structure and function.

**Outcome 3**

Test and modify the program to provide a solution to a problem.

**Evidence requirements**

- 3.1 A procedure for testing the program is described and documented, and ensures the program meets the design specifications.
- 3.2 Testing follows the documented procedure and the program is modified as required to meet design specifications.

Range all changes are documented, including the reasons for them.

- 3.3 A print-out of the program code and comments verifies readability and presentation.
- 3.4 Operation of the program verifies that the program meets the design specification and provides a solution to the problem.

**Outcome 4**

Create simple end-user documentation for the program.

**Evidence requirements**

- 4.1 A simple end-user document is created to facilitate use of the program.

Range information supplied includes a short description of the purpose of the program, and instructions on how to access and use the program.

<b>Planned review date</b>	31 December 2016
----------------------------	------------------

#### Status information and last date for assessment for superseded versions

Process	Version	Date	Last Date for Assessment
Registration	1	23 October 2002	31 December 2013
Revision	2	16 July 2004	31 December 2013
Review	3	22 May 2009	31 December 2013
Revision	4	18 February 2011	31 December 2015
Rollover and Revision	5	19 September 2013	N/A

<b>Consent and Moderation Requirements (CMR) reference</b>	0226
--	------

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

#### Please note

Providers must be granted consent to assess against standards (accredited) by NZQA, before they can report credits from assessment against unit standards or deliver courses of study leading to that assessment.

Industry Training Organisations must be granted consent to assess against standards by NZQA before they can register credits from assessment against unit standards.

Providers and Industry Training Organisations, which have been granted consent and which are assessing against unit standards must engage with the moderation system that applies to those standards.

Requirements for consent to assess and an outline of the moderation system that applies to this standard are outlined in the Consent and Moderation Requirements (CMR). The CMR also includes useful information about special requirements for organisations wishing to develop education and training programmes, such as minimum qualifications for tutors and assessors, and special resource requirements.

#### Comments on this unit standard

Please contact NZQA National Qualifications Services [nqs@nzqa.govt.nz](mailto:nqs@nzqa.govt.nz) if you wish to suggest changes to the content of this unit standard.