Achievement Standard

Subject Reference	Digital Technologies 2.45
Title	Construct a plan for an advanced computer program for a specified task

Level	2	**Credits**	3	**Assessment**	Internal

Subfield Technology

Domain Digital Technologies

Status	Registered	**Status date**	17 November 2011
Planned review date	31 December 2019	**Date version published**	17 November 2016

This achievement standard requires constructing a plan for an advanced computer program for a specified task.

Achievement Criteria

Achievement	Achievement with Merit	Achievement with Excellence
• Construct a plan for an advanced computer program for a specified task.	• Skilfully construct a plan for an advanced computer program for a specified task.	• Efficiently construct a plan for an advanced computer program for a specified task.

Explanatory Notes

1 This achievement standard is derived from Level 7 of the Technology learning area in *The New Zealand Curriculum*, Learning Media, Ministry of Education, 2007; and is related to the material in the *Teaching and Learning Guide for Technology*, Ministry of Education at http://seniorsecondary.tki.org.nz.

Further information can be found at http://www.technology.tki.org.nz/.

Appropriate reference information is available in *Safety and Technology Education: A Guidance Manual for New Zealand Schools,* Ministry of Education at http://technology.tki.org.nz/Curriculum-support/Safety-and-Technology-Education, and the Health and Safety at Work Act 2015.

This standard is also derived from *Te Marautanga o Aotearoa*. For details of *Te Marautanga o Aotearoa* achievement objectives to which this standard relates, see the Papa Whakaako for the relevant learning area.

2   *Construct a plan for an advanced computer program for a specified task* involves:
- specifying variables, their scopes and data types
- specifying an indexed data structure
- specifying a modular structure for the program, including details of the procedural structures of the modules
- specifying a set of expected input cases for testing the program.

Skilfully construct a plan for an advanced computer program for a specified task involves:
- independently constructing the plan
- specifying well-chosen scopes for the variables
- specifying well-chosen parameters for the modules
- specifying a set of expected and boundary input cases for testing the program.

Efficiently construct a plan for an advanced computer program for a specified task involves:
- specifying modules (including their procedural structures) that constitute a well-structured logical decomposition of the task
- specifying variables, constants, and derived values effectively so as to maximise the flexibility and robustness of the plan
- specifying a comprehensive set of expected, boundary and exceptional input cases for testing the program.

3   A *plan* refers to a coherent document that communicates the structure and function of a computer program. The plan must be developed in the context of a target programming language. The plan may be communicated using a range of tools. These may include but are not limited to: natural language (as long as it is unambiguous and precise), diagrams (including screen layouts), and pseudocode. The plan itself should not be expressed in program code.

4   *An advanced computer program* must have a modular structure, an indexed data structure (e.g. array or list), input and output, and procedural structures that combine sequential, conditional and iterative structures.

5   A plan for a modular structure is a collection of named modules where each module specifies a procedural structure for a sub-task. At least the top-level module (and possibly others) must contain calls to other modules. The modules should include parameters as needed.

6   *A specified task* refers to a set task which requires the development of an advanced computer program to resolve. The task must be of sufficient rigour to allow the student to meet the standard and needs to be agreed prior to the plan being constructed. It may be teacher-given or developed in negotiation with the student.

7   The scope of a variable may be global (accessible from all modules) or limited to a single module. A well-chosen scope matches the way the variable is used.

8   Well-chosen parameters for modules are those where the number and types of parameters are decided in the context of the module's sub-task.

9 Constants should be used as required when a value never changes. Derived values are returned properties or are calculated from other values. Examples include but are not limited to: the length of an array or string; area which is calculated from the width and height of a rectangle; and the mid-point of a graphics object which is calculated from its width and height.

10 In a well-structured logical decomposition of the task, each module will have a clear and well-defined purpose within the context of the task. Interaction between modules will be minimised, modules will be reused rather than duplicated, and the procedural structure of each module will be efficient.

11 Conditions of Assessment related to this achievement standard can be found at http://ncea.tki.org.nz/Resources-for-Internally-Assessed-Achievement-Standards.

Quality Assurance

1 Providers and Industry Training Organisations must have been granted consent to assess by NZQA before they can register credits from assessment against achievement standards.

2 Organisations with consent to assess and Industry Training Organisations assessing against achievement standards must engage with the moderation system that applies to those achievement standards.

Consent and Moderation Requirements (CMR) reference 0233