

Achievement Standard

Subject Reference	Digital Technologies 2.46		
Title	Construct 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 2014	Date version published	17 November 2011

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

Achievement Criteria

Achievement	Achievement with Merit	Achievement with Excellence
<ul style="list-style-type: none"> Construct an advanced computer program for a specified task. 	<ul style="list-style-type: none"> Skilfully construct an advanced computer program for a specified task. 	<ul style="list-style-type: none"> Efficiently construct an advanced computer program for a specified task.

Explanatory Notes

- This achievement standard is derived from the Level 7 achievement objectives from 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, 2010 at <http://seniorsecondary.tki.org.nz>.

Appropriate reference information is available in *Safety and Technology Education: A Guidance Manual for New Zealand Schools*, Learning Media, Ministry of Education, 1998; and the Health and Safety in Employment Act 1992.

Further information can be found at <http://www.techlink.org.nz>.

- 2 *Construct an advanced computer program for a specified task* involves:
- implementing a plan for an advanced program in a suitable programming language
 - setting out the program code clearly and documenting the program with comments
 - testing and debugging the program to ensure it works on a sample of expected input cases.

Skilfully construct an advanced computer program for a specified task involves:

- independently implementing a plan for an advanced program in a suitable programming language that uses well-chosen scopes for variables, and well-chosen parameters for modules
- documenting the program with variable and module names and comments that accurately describe code function and behaviour
- testing and debugging the program in an organised way to ensure it works on inputs that include both expected and boundary cases.

Efficiently construct an advanced computer program for a specified task involves:

- constructing an advanced program where the modules (including their procedural structures) constitute a well-structured logical decomposition of the task
- using variables, constants, and derived values effectively so as to increase the flexibility and robustness of the program
- setting out the program code concisely and documenting the program with comments that explain and justify decisions
- comprehensively testing and debugging the program in an organised and time-effective way to ensure the program is correct on expected, boundary and invalid inputs.

- 3 It is preferable for the programming language for this standard to be text-based. Any language chosen must support indexed data structures, modules with parameters, global and local scope of variables and good comment/document facilities.
- 4 *An advanced computer program* must have a modular structure, an indexed data structure (eg array or list), input and output, and procedural structures that combine sequential, conditional and iterative structures.
- 5 A program with a modular structure contains a collection of named modules (procedures, functions, methods, or subroutines) where each module implements 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 program being constructed. It may be teacher-given or developed in negotiation with the student. It is expected that most students constructing a program for this standard will have developed a plan for an advanced computer program for this task. If this is not the case, the teacher should provide the student with an abstract plan to guide their program development but one which cannot be directly transcribed to the programming language.

- 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://www.tki.org.nz/e/community/ncea/conditions-assessment.php>.
-

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