

Achievement Standard

Subject Reference Digital Technologies 1.44

Title Demonstrate understanding of basic concepts from computer science

Level 1 **Credits** 3 **Assessment** External

Subfield Technology

Domain Digital Technologies

Status Registered **Status date** 20 January 2011

Planned review date 31 December 2014 **Date version published** 20 January 2011

This achievement standard requires demonstrating understanding of basic concepts from computer science.

Achievement Criteria

Achievement	Achievement with Merit	Achievement with Excellence
<ul style="list-style-type: none"> Demonstrate understanding of basic concepts from computer science. 	<ul style="list-style-type: none"> Demonstrate in-depth understanding of basic concepts from computer science. 	<ul style="list-style-type: none"> Demonstrate comprehensive understanding of basic concepts from computer science.

Explanatory Notes

- This achievement standard is derived from the Level 6 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>.

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

- Demonstrate understanding of basic concepts from computer science* involves:
 - describing the key characteristics and roles of algorithms, programs and informal instructions
 - describing an algorithm for a task, showing understanding of the kinds of steps that can be in an algorithm, and determining the cost of an algorithm for a problem of a particular size

- describing the role and characteristics of programming languages, including the different roles and characteristics of high level languages and low level (or machine) languages, and the function of a compiler
- describing the role of a user interface and factors that contribute to its usability.

Demonstrate in-depth understanding of basic concepts from computer science involves:

- explaining how algorithms are distinct from related concepts such as programs and informal instructions
- showing understanding of the way steps in an algorithm for a task can be combined in sequential, conditional, and iterative structures and determining the cost of an iterative algorithm for a problem of size n
- explaining how the characteristics of programming languages, including the different characteristics of high level and low level (or machine) languages, are important for their roles
- explaining the need for programs to translate between high and low level languages
- explaining how different factors of a user interface contribute to its usability.

Demonstrate comprehensive understanding of basic computer science concepts from computer science involves:

- comparing and contrasting the concepts of algorithms, programs, and informal instructions
- determining and comparing the costs of two different iterative algorithms for the same problem of size n
- comparing and contrasting high level and low level (or machine) languages, and explaining different ways in which programs in a high level programming language are translated into a machine language
- discussing how different factors of a user interface contribute to its usability by comparing and contrasting related interfaces.

- 3 The *basic concepts from computer science* are: the concept of an algorithm; the concept of a programming language; and the concept of a user interface and its usability.
- 4 An algorithm is a precise unambiguous specification of how to accomplish some computational task in a finite number of well-defined steps. An algorithm is distinct from a computer program. An algorithm has a cost (the number of steps it will perform) for a task. Different algorithms for the same task may have different costs.
- 5 A programming language is a precise, formal language for writing programs that can be run on a computer; it is distinct from pseudocode and natural language. There are different levels of programming languages; programs can be translated from high level to low level (or machine) languages.
- 6 A user interface is the part of a computer or electronic system that a human user interacts with to control the system. The usability of an interface is the key characteristic for evaluating an interface.
- 7 Assessment Specifications for this achievement standard can be accessed through the Technology Resources page found at <http://www.nzqa.govt.nz/ncea/resources>.

Quality Assurance

- 1 Providers and Industry Training Organisations must be accredited by NZQA before they can register credits from assessment against achievement standards.
- 2 Accredited providers and Industry Training Organisations assessing against achievement standards must engage with the moderation system that applies to those achievement standards.

Accreditation and Moderation Action Plan (AMAP) reference

0233

Achievement Standard

Subject Reference Digital Technologies 1.45

Title Construct a plan for a basic computer program for a specified task

Level 1 **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 a plan for a basic computer program for a specified task.

Achievement Criteria

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

Explanatory Notes

- This achievement standard is derived from the Level 6 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>.

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

- Construct a plan for a basic computer program for a specified task* involves:
 - specifying variables and their data types
 - specifying a procedural structure that combines actions, conditions and control structures
 - specifying a set of test cases with expected inputs for testing the program.

Skilfully construct a plan for a basic computer program for a specified task involves:

- independently constructing the plan
- specifying a procedural structure with well-chosen actions, conditions and control structures
- specifying a set of test cases with expected and boundary inputs for testing the program.

Efficiently construct a plan for a basic computer program for a specified task involves:

- constructing a flexible and robust plan
- specifying an effective procedural structure that constitutes a well-structured logical solution to the task
- specifying a comprehensive set of test cases with expected, boundary and invalid input 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 A *basic computer program* is a program written in a programming language, that uses:
 - variables involving at least two types of information (e.g. numeric, characters, text) and assignment
 - sequence, selection and iteration control structures
 - predefined actions (e.g. predefined methods, functions, or procedures)
 - input from a user, sensors, or other external source.
- 5 A *specified task* refers to a set task which requires the development of a basic 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.
- 6 Actions are the atomic steps of the plan. Conditions are logical expressions within conditional and iterative structures that control the choice or repetitions.
- 7 Well-chosen actions, conditions and control structures are those where the resulting sequence of actions correctly performs the task and has no unintended behaviour or consequences.
- 8 Ways of making a plan more flexible and robust may include but are not limited to: checking input data for validity; correctly handling expected, boundary and invalid inputs; using constants and variables of appropriate data types; and using constants, variables and derived values in place of literals.
- 9 An effective procedural structure is one where each individual control structure has a clear and well defined purpose within the structure of the plan, and which has no unnecessary duplication or repetition.
- 10 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

Achievement Standard

Subject Reference Digital Technologies 1.46

Title Construct a basic computer program for a specified task

Level 1 **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 a basic computer program for a specified task.

Achievement Criteria

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

Explanatory Notes

- This achievement standard is derived from the Level 6 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>.

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

- Construct a basic computer program for a specified task involves:
 - implementing a plan for a basic 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 that it works on a sample of expected inputs.

Skilfully construct a basic computer program for a specified task involves:

- independently implementing the plan for a basic program in a suitable programming language that uses a procedural structure with well-chosen actions, conditions and control structures
- documenting the program with variable names and comments that accurately describe code function and behaviour
- testing and debugging the program in an organised way to ensure that it works on expected and boundary inputs.

Efficiently construct a basic computer program for a specified task involves:

- constructing a basic program which uses actions, conditions and control structures effectively to increase the flexibility and robustness of the program
- using an effective procedural structure that results in a well-structured, logical solution to the task
- setting out the program code concisely and documenting the program with succinct 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 The programming language could be graphical, drag-and-drop or text based. The language chosen must support the required data types and procedural structures, and good comment facilities.
- 4 A *basic computer program* is a program written in a programming language, that uses:
 - variables involving at least two types of information (e.g. numeric, characters, text) and assignment
 - sequence, selection and iteration control structures
 - predefined actions (e.g. predefined methods, functions, or procedures)
 - input from a user, sensors, or other external source.
- 5 A *specified task* refers to a set task which requires the development of a basic 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 a basic 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.
- 6 Actions are the atomic statements of the program (including the use of predefined methods, functions, or procedures). Conditions are logical expressions within conditional and iterative structures that control the choice or repetitions.
- 7 Well-chosen actions, conditions and control structures are those where the resulting sequence of actions correctly performs the task and has no unintended behaviour or consequences.

- 8 Ways of making a program more flexible and robust must include using actions, conditions and control structures effectively and may also include but are not limited to: checking input data for validity; correctly handling expected, boundary and invalid inputs; and using constants, variables and derived values in place of literals.
 - 9 An effective procedural structure is one where each individual control structure has a clear and well defined purpose within the structure of the program, and which has no unnecessary duplication or repetition.
 - 10 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

Achievement Standard

Subject Reference Digital Technologies 2.44

Title Demonstrate understanding of advanced concepts from computer science

Level 2 **Credits** 4 **Assessment** External

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 demonstrating understanding of advanced concepts from computer science.

Achievement Criteria

Achievement	Achievement with Merit	Achievement with Excellence
<ul style="list-style-type: none"> Demonstrate understanding of advanced concepts from computer science. 	<ul style="list-style-type: none"> Demonstrate in-depth understanding of advanced concepts from computer science. 	<ul style="list-style-type: none"> Demonstrate comprehensive understanding of advanced concepts from computer science.

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 *Demonstrate understanding of advanced concepts from computer science* involves:
- describing ways in which different types of data can be represented using bits
 - describing the concept of encoding information using compression coding, error control coding, and encryption; and typical uses of encoded information
 - providing examples from human-computer interfaces that illustrate usability heuristics.

Demonstrate in-depth understanding of advanced concepts from computer science involves:

- comparing and contrasting different ways in which different types of data can be represented using bits and discussing the implications
- discussing how a widely used technology is enabled by one or more of compression coding, error control coding, and encryption
- evaluating a given human-computer interface in terms of usability heuristics.

Demonstrate comprehensive understanding of advanced concepts from computer science involves:

- evaluating a widely used system for compression coding, error control coding, or encryption
- suggesting improvements to a given human-computer interface based on an evaluation in terms of usability heuristics.

- 3 *Advanced concepts from computer science* are the concepts of data representations, encoding, and usability heuristics.
- 4 Data representations may include binary representations for signed and unsigned integers, real numbers, characters, text, colours, sound, and images.
- 5 Assessment Specifications for this achievement standard can be accessed through the Technology Resources page found at <http://www.nzqa.govt.nz/qualifications-standards/qualifications/ncea/ncea-subject-resources/>.

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

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 2014 **Date version published** 17 November 2011

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

- 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 (eg 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://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

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

Achievement Standard

Subject Reference Digital Technologies 3.44

Title Demonstrate understanding of areas of computer science

Level 3 **Credits** 4 **Assessment** External

Subfield Technology

Domain Digital Technologies

Status Registered **Status date** 4 December 2012

Planned review date 31 December 2016 **Date version published** 4 December 2012

This achievement standard involves demonstrating understanding of areas of computer science.

Achievement Criteria

Achievement	Achievement with Merit	Achievement with Excellence
<ul style="list-style-type: none"> Demonstrate understanding of areas of computer science. 	<ul style="list-style-type: none"> Demonstrate in-depth understanding of areas of computer science. 	<ul style="list-style-type: none"> Demonstrate comprehensive understanding of areas of computer science.

Explanatory Notes

- This achievement standard is derived from the Level 8 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, 2012, at <http://seniorsecondary.tki.org.nz/>.

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

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

- Demonstrate understanding of areas of computer science* involves:
 - describing key problems that are addressed in selected areas of computer science
 - describing examples of practical applications of selected areas to demonstrate the use of key algorithms and/or techniques from these areas.

Demonstrate in-depth understanding of areas of computer science involves:

- explaining how key algorithms or techniques are applied in selected areas
- explaining examples of practical applications of selected areas to demonstrate the use of key algorithms and/or techniques from these areas.

Demonstrate comprehensive understanding of areas of computer science involves:

- discussing examples of practical applications of selected areas to demonstrate the use of key algorithms and/or techniques from these areas
- evaluating the effectiveness of algorithms, techniques, or applications from selected areas.

3 *Areas of Computer Science* will be selected from:

- formal languages
- network communication protocols
- complexity and tractability
- intelligent systems
- software engineering
- graphics and visual computing.

4 Assessment Specifications for this achievement standard can be accessed through the Technology Resources page found at <http://www.nzqa.govt.nz/qualifications-standards/qualifications/ncea/ncea-subject-resources/>.

Replacement Information

This achievement standard, AS91632, AS91634, and AS91641 replaced AS90684.

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

Achievement Standard

Subject Reference	Digital Technologies 3.46		
Title	Develop a complex computer program for a specified task		
Level	3	Credits	6
		Assessment	Internal
Subfield	Technology		
Domain	Digital Technologies		
Status	Registered	Status date	4 December 2012
Planned review date	31 December 2016	Date version published	4 December 2012

This achievement standard involves developing a complex computer program for a specified task.

Achievement Criteria

Achievement	Achievement with Merit	Achievement with Excellence
<ul style="list-style-type: none"> Develop a complex computer program for a specified task. 	<ul style="list-style-type: none"> Skilfully develop a complex computer program for a specified task. 	<ul style="list-style-type: none"> Efficiently develop a complex computer program for a specified task.

Explanatory Notes

- This achievement standard is derived from the Level 8 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, 2012, at <http://seniorsecondary.tki.org.nz/>.

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

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

- Develop a complex computer program for a specified task* involves:
 - designing and implementing a program that includes variables, an indexed data structure, and a modular structure including details of the procedural structures of the modules
 - including a working graphical user interface with different sources of event generating components and event handling, and using classes and objects to encapsulate data and methods

- 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 develop a complex computer program for a specified task involves:

- using well-chosen modular and procedural structures, scope and encapsulation for data and methods, graphical user interface and event handling mechanisms
- documenting the program with variable and module names and comments that accurately describe code function and behaviour.
- following a disciplined design and implementation process, with documented cycles of incremental development and comprehensive testing process, to ensure that the program works on inputs that include both expected and boundary cases.

Efficiently develop a complex computer program for a specified task involves:

- ensuring that the overall modular and procedural design, graphical user interface, and event handling design, are a well-structured, logical decomposition of the task, and that the program is flexible and robust
- 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 input cases.

- 3 The programming language for this standard must be a text-based Object-oriented programming language that supports graphical user interfaces (GUIs) and event based programming.
- 4 The platform for the program could be a personal computer, a mobile device, or a web browser. The program must involve graphical elements such as widgets, images, and/or shapes, and must respond to event based input from the user.
- 5 A *complex computer program* is one that has a modular structure; an indexed data structure (eg array or list); input and output; procedural structures that combine sequential, conditional, and iterative structures; a graphical user interface and event handling; and that includes classes and objects. Inheritance is not required.
- 6 Any code that is auto-generated by an Integrated Development Environment (IDE) cannot be used as evidence of student code.
- 7 A *specified task* refers to a set task which requires the development of a complex 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 developed. It may be teacher-given or developed in negotiation with the student.
- 8 Conditions of Assessment related to this achievement standard can be found at <http://ncea.tki.org.nz/>.

Replacement Information

This achievement standard, AS91633, AS91635, and AS91642 replaced unit standard 13402 and AS90685.

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