

DIGITAL TECHNOLOGIES: CONSTRUCT A SOFTWARE PROGRAM

Construct a software program focuses on constructing a computer program for a specified task including testing and debugging the program to ensure the program works correctly.

Initially students learn to construct basic computer programs in any programming language (drag-and-drop language, specialised programming language, or a general purpose programming language) that include:

- variables, assignment, predefined actions, expressions, and sequence, selection, and iteration control structures; and
- obtains and uses input from a user, sensors, or other external source.

Students progress to constructing complex computer programs using a text based programming language

	LEVEL 6	LEVEL 7	LEVEL 8
LO	<i>Demonstrate ability to construct a basic software program</i>	<i>Demonstrate ability to construct an advanced software program</i>	<i>Develop a complex computer program for a specified task</i>
TEACHER GUIDANCE	<p>To support students to develop an ability to construct a basic software program at level 6, teachers could:</p> <ul style="list-style-type: none"> • Guide students to independently implement a plan for a basic program in a suitable programming language (drag-and-drop language, specialised programming language, or a general purpose programming language) that uses a procedural structure with well-chosen actions, conditions and control structures that ensures the program is flexible and robust • Guide students on how to set out program code concisely, and document programs with variable names and succinct comments that accurately explain and justify code function and behaviours • Guide students on how to comprehensively test and debug programs in an organised time effective way to ensure that they work on expected, boundary and invalid inputs. 	<p>To support students to develop an ability to construct an advanced software program at level 7, teachers could:</p> <ul style="list-style-type: none"> • Guide students on how to independently implement a plan to construct advanced programs, in suitable programming language, where the modules (including their procedural structures) constitute a well-structured logical decomposition of the tasks • Guide students on how to use variables, constants, and derived values effectively to increase the flexibility and robustness of programs • Guide students on how to set out program code clearly and document programs with variables and module names, and include comments that explain and justify code functions and behaviours • Guide students on how to comprehensively test and debug programs in organised and time effective ways to ensure that programs are correct on expected, boundary's and invalid inputs. 	<p>To support students to develop a complex software program, at level 8, teachers could:</p> <ul style="list-style-type: none"> • Ensure students understand the requirement at this level that the programming language must be a text-based programming language and have an appropriate IDE that includes debugging tools. • Guide students on how to use an Integrated Development Environment (IDE) to develop code following a disciplined development process with cycles of incremental development and testing. • Guide students on how to construct a complex computer program in a text-based programming language that supports object-orientated structuring. • Guide students on how to follow accepted testing and debugging practices using IDE debugging tools to test and debug a program to ensure it works for expected, boundary, and exceptional cases. • Provide opportunities for students to practice using an appropriate IDE to develop code following a disciplined development process with cycles of incremental development and testing. Provide opportunities for students to practice constructing and testing complex computer programs. • Ensure students understand the requirement at this level that a complex program is a program written in a text-based programming language that interacts with a user, includes variables, assignment, predefined actions, expressions, includes sequence, selection, iteration control structures, includes programmer defined methods/ functions/etc with parameters and/or return values, includes calls to the methods/functions/etc, uses structured data, including sequential data (arrays, lists, etc) and compound data (records, objects, tuples etc), uses and updates persistent data in files or databases, has structuring of the methods/functions/etc and data (eg, Classes, modules, encapsulated data structures, packages, etc). • Ensure students have a specified task that requires the development of a complex program to resolve the task. The task can be teacher-given or developed in negotiation with the student. Ensure the task is large enough to justify decomposition of a program into multiple classes (or other high level modules).
INDICATORS	<p>Students can:</p> <ul style="list-style-type: none"> • write a program with sequence, selection, and iteration control structures • write a program with multiple data types, iteration control structures nested inside other iteration control structures, and structures in which complex logical conditions are expressed economically • comprehensively test and debug the program in an organised and time effective way to ensure the program is correct on all inputs (including expected, exceptional, out-of-range, boundary, and invalid inputs). 	<p>Students can:</p> <ul style="list-style-type: none"> • independently implement a plan for an advanced program in a suitable programming language (preferably a text-based programming language) • construct an advanced program where the modules (including their procedural structures) constitute a well-structured logical decomposition of the task • use variables, constants, and derived values effectively so as to increase the flexibility and robustness of the program • set out the program code clearly and concisely and document the program with comments that explain and justify decisions • comprehensively test and debug the program in an organised and time effective way to ensure the program is correct on expected, boundary and invalid inputs. 	<p>Students can:</p> <ul style="list-style-type: none"> • use an appropriate IDE to develop code and use the IDE debugging tools effectively to identify logic errors and correct a program • follow a disciplined and planned development process with documented cycles of incremental development and comprehensive testing at each cycle to construct a correctly working program • follow accepted debugging practices by interpreting syntax and runtime error messages to identify the underlying errors and correct a program • follow accepted testing and debugging practices for systematically applying test cases and using tracing/debugging statements to identify logic errors and correct a program. • write a computer program in a text-based programming language that includes commented, programmer defined methods/functions/ etc with parameters and/or return values, has structuring of the methods/functions/etc and data (eg, classes, modules, encapsulated data structures, packages, etc), and has well-designed algorithmic structures for the individual methods/functions/etc. • write a program that includes functions/methods/procedures that are passed compound data structures (arrays, lists, objects, etc) and modify their contents, and has a well-designed decomposition into functions/methods with well-chosen parameters and has a well-designed structuring of data and methods/functions/etc into classes (or modules, packages, etc) • include explanatory comments and identifiers that support maintainability (including informative comments on functions/methods/procedures) • test their program to ensure it works correctly.
AS	<p>AS91076 Digital Technologies 1.46 <i>Construct a basic computer program for a specified task</i></p>	<p>AS91373 Digital Technologies 2.46 <i>Construct an advanced computer program for a specified task</i></p>	<p>AS91637 Digital Technologies 3.46 <i>Develop a complex computer program for a specified task</i></p>
	Level 1 Digital Technologies standards	Level 2 Digital Technologies standards & assessment	Level 3 Technology achievement standards & assessment resources DRAFT