



Australian Government

Department of Education, Employment and Workplace Relations

ICAPRG301A Apply introductory programming techniques

Release: 1

ICAPRG301A Apply introductory programming techniques

Modification History

Release	Comments
Release 1	This Unit first released with <i>ICA11 Information and Communications Technology Training Package version 1.0</i>

Unit Descriptor

This unit describes the performance outcomes, skills and knowledge required to undertake introductory programming tasks in the development of a game or application.

This unit addresses the knowledge, processes and techniques necessary to develop skills to create simple applications or games.

Application of the Unit

This unit applies to people with responsibility for creating applications or games and includes creating code, using programming standards, testing and debugging.

Licensing/Regulatory Information

No licensing, legislative, regulatory or certification requirements apply to this unit at the time of endorsement but users should confirm requirements with the relevant federal, state or territory authority.

Pre-Requisites

Not applicable.

Employability Skills Information

This unit contains employability skills.

Elements and Performance Criteria Pre-Content

Element	Performance Criteria
<i>Elements describe the essential outcomes of a unit of competency.</i>	<i>Performance criteria describe the performance needed to demonstrate achievement of the element. Where bold italicised text is used, further information is detailed in the required skills and knowledge section and the range statement. Assessment of performance is to be consistent with the evidence guide.</i>

Elements and Performance Criteria

1. Apply language syntax and layout	1.1 Apply basic language syntax rules 1.2 Use language data types, operators and expressions to create clear and concise code 1.3 Apply variables and variable scope 1.4 Use library functions in a program 1.5 Use commenting to create clear meaning to code
2. Apply control structures	2.1 Apply language syntax for sequence, selection and iteration constructs 2.2 Use logical operators to create expressions for use in selection and iteration constructs
3. Code using standard algorithms	3.1 Develop algorithms that use the sequence, selection and iteration constructs 3.2 Create and use arrays 3.3 Code standard sequential access algorithms for reading and writing text files, including end-of-file detection loops 3.4 Apply string manipulation
4. Test code	4.1 Use debugging techniques to trace code execution and examine variable contents to detect and correct errors 4.2 Create and conduct simple tests to confirm code meets design specification 4.3 Document the tests performed and results achieved
5. Create an application or game	5.1 Design an algorithm in response to basic program specifications 5.2 Develop the application or game to meet the program specification 5.3 Test and confirm that application or game meets the initial specifications

Required Skills and Knowledge

This section describes the skills and knowledge required for this unit.

Required skills

- analytical skills to determine the requirements of the specification
- communication skills to:
 - document the code
 - understand the client requirements
- learning skills to research encountered problems independently
- literacy skills to create program code that uses correct syntax
- planning and organisational skills to complete the program within the required timeframe
- problem-solving skills to translate the required outcomes of the specification into correct program code
- technical skills to apply programming concepts.

Required knowledge

- programming language syntax and command structure
- small-sized application or game development.

Evidence Guide

The evidence guide provides advice on assessment and must be read in conjunction with the performance criteria, required skills and knowledge, range statement and the Assessment Guidelines for the Training Package.

Overview of assessment	
Critical aspects for assessment and evidence required to demonstrate competency in this unit	Evidence of the ability to: • apply programming language syntax, sequence, selection and iteration control structures to the development of an application or game • produce an application or game that is designed and built from a provided program specification • confirm that the created application or game meets the original program specifications and obtain user sign-off for completed program.
Context of and specific resources for assessment	Assessment must ensure access to: • requirement documents • site documents • software development tools currently used in industry, such as compiler and IDE • appropriate learning and assessment support when required • modified equipment for people with special needs.
Method of assessment	A range of assessment methods should be used to assess practical skills and knowledge. The following examples are appropriate for this unit: • direct observation of candidate programming and debugging code • verbal or written questioning to assess candidate's knowledge of language syntax and programming constructs • review of programs to assess candidate's knowledge of language syntax and programming constructs • review of an application or game developed by candidate to meet a specification.
Guidance information for assessment	Holistic assessment with other units relevant to the industry sector, workplace and job role is recommended, where appropriate. Assessment processes and techniques must be culturally appropriate, and suitable to the communication skill level, language, literacy and numeracy capacity of the candidate and the work being performed. Indigenous people and other people from a non-English speaking

	background may need additional support. In cases where practical assessment is used it should be combined with targeted questioning to assess required knowledge.
--	---

Range Statement

The range statement relates to the unit of competency as a whole. It allows for different work environments and situations that may affect performance. Bold italicised wording, if used in the performance criteria, is detailed below. Essential operating conditions that may be present with training and assessment (depending on the work situation, needs of the candidate, accessibility of the item, and local industry and regional contexts) may also be included.

Language may include:	• C • C++ • Java • VB.
Variable scope may include:	• global • local • local to code block.
Library functions may include:	• getch() • printf, cout • rand() • SDL_BlitterSurface() • SDL_PollEvent().
Logical operators may include:	• AND • OR • NOT.
Arrays may include:	• one-dimensional arrays: • declare • initialise • loop through array elements • use • 2-D arrays.
String manipulation may include:	• converting strings to numbers • searching a string for tokens • splitting a string based on tokens • stringing concatenation • sub-strings.
Debugging techniques may include:	• printing variable information to screen or log • using debugging tools provided by the integrated development environment (IDE).

Unit Sector(s)

Programming and software development