



Australian Government

Department of Education, Employment and Workplace Relations

ICAPRG414A Apply introductory programming skills in another language

Release: 1

ICAPRG414A Apply introductory programming skills in another language

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 programming tasks using a procedural approach to programming. An object-oriented language may be used in this approach.

Application of the Unit

This unit applies to individuals working in programming, development and technical roles.

They may work as programmers, software developers or may be IT staff with responsibility for undertaking programming activities, including writing, maintaining and updating programs, defining data and file handling.

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 basic 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 Use appropriate language syntax for sequence, selection and iteration constructs
2. Code using data structures	2.1 Demonstrate understanding of data structures 2.2 Write code to create and manipulate arrays 2.3 Design, define and use data structures
3. Code using standard algorithms	3.1 Create sequential search, binary search, insertion and deletion algorithms to operate on arrays 3.2 Code standard sequential access algorithms and random access algorithms
4. Debug code	4.1 Use stand-alone debugging tools or tools provided by integrated development environment (IDE) to debug code 4.2 Use a debugger to trace code execution and examine variable contents to detect and correct errors
5. Document activities	5.1 Follow organisational guidelines for developing maintainable code and adhere to the provided coding standard when documenting activities 5.2 Apply internal documentation suitable for use by peers to all code created and use documentation tools available in the target language when documenting activities
6. Test code	6.1 Design and document tests 6.2 Capture and record test results

Required Skills and Knowledge

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

Required skills

- literacy skills to:
 - read and interpret:
 - code standards and organisational guidelines
 - program specifications
 - write internal documentation
- problem-solving skills to develop solutions for applications
- technical skills to use integrated development environment.

Required knowledge

- basic knowledge of:
 - documentation techniques
 - programming techniques
 - testing techniques
 - small-size application 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: - perform programming coding to create, debug and test medium-size applications - generate design and code documentation - test and confirm that created application meets program specifications.
Context of and specific resources for assessment	Assessment must ensure access to: - IDE for the determined language - specific tools and licences, depending on particular platform and language - 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: - evaluation of candidate's application code, test and documentation - verbal or written questioning of candidate on key programming concepts, such as: - syntax and language features - aggregate data types (such as collections, lists, hash tables, arrays and queues) that are appropriate to the language studied - using an IDE - debugging code.
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.

<i>Language</i> may include:	• C++ • C# • Java • JavaScript • Visual Basic (VB) • VBScript.
<i>Constructs</i> may include:	• iterations or loops • nested control structures • selection statements: • if • switch.
<i>Data structures</i> may include:	• arrays • collections • dictionaries • lists • maps • sets.
<i>Arrays</i> may include:	• multi-dimensional arrays • one-dimensional arrays.
<i>Integrated development environment</i> may include:	• C++Builder • CodeLite • Eclipse • JavaBuilder • Microsoft Visual Studio • NetBeans • Xcode.
<i>Organisational guidelines</i> may include:	• communication methods • content of emails • dispute resolution • document procedures • downloading information and accessing particular websites • financial control mechanisms • opening mail with attachments

	• personal use of emails and internet access • templates • virus risk.
Documentation may include:	• audit trails • International Organization for Standardization (ISO), International Electrotechnical Commission (IEC), and Australian Standards (AS) standards • naming standards • project management templates • report writing • version control.

Unit Sector(s)

Programming and software development