



Australian Government

Department of Education, Employment and Workplace Relations

ICAPRG523A Apply advanced programming skills in another language

Release: 1

ICAPRG523A Apply advanced 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 advanced programming tasks using a selected choice of another programming language. The second language may be an object-oriented language.

Application of the Unit

This unit applies to programmers who are required to develop program code.

Programming languages form the underpinning software infrastructure of all computer data processing. It is important that the stages of the development methodology are followed within the scope of any project or scenario, and the relevant supporting documentation produced.

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. Code using advanced data structures	1.1 Design dynamic data structures 1.2 Implement and use dynamic data structures, including double-linked lists and binary trees for coding
2. Code using advanced algorithms	2.1 Code using hashing techniques 2.2 Consider and record the advantages and disadvantages of at least three sorting algorithms 2.3 Code at least one sorting algorithm 2.4 Code advanced searching techniques for use with complex data structures
3. Write simple multi-process application	3.1 Demonstrate use of the features of the language that enable inter-process communication through at least one mechanism 3.2 Demonstrate use of features of the language that allow for operating system 'signals' to be captured and responded to
4. Use third-party libraries	4.1 Use a third-party library in the construction of an application 4.2 Reference third-party documentation 4.3 Use procedural techniques to write an application to work within a graphical user interface (GUI) environment
5. Debug code	5.1 Use integrated development environment (IDE) debugging facilities or a stand-alone debugger 5.2 Use program debugging techniques to detect and resolve errors of syntactical, logical and design origin
6. Document activities	6.1 Demonstrate use of source code version control 6.2 Demonstrate adherence to guidelines for developing maintainable code and to a set of provided coding standards 6.3 Apply suitable internal documentation to all code created using documentation tools available in the target language
7. Test code	7.1 Design and document tests to be undertaken 7.2 Undertake limited testing of produced code to ensure it complies with program specification 7.3 Capture test results
8. Create an application	8.1 Develop a solution from a program specification design document 8.2 Design the algorithm, and construct and test applications in response to a problem description and language

Required Skills and Knowledge

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

Required skills

- analytical skills to read and interpret program specifications
- literacy skills to write technical documentation
- problem-solving skills to solve bugs in program code
- technical skills to:
 - debug, program and test code in specified language
 - use an IDE
 - work with version control.

Required knowledge

- detailed knowledge of:
 - range of programming languages
 - languages in a GUI environment
 - techniques for developing large scale applications
 - procedures for programming with complex data structures
 - procedures for using third-party supplied library for standard programming features.

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: • design application programs using a range of tools to support programming activities • produce design documentation for these programs • build application programs according to the design • document these programs • conduct testing procedures on these programs to ensure validity • analyse test results to determine program's accuracy.
Context of and specific resources for assessment	Assessment must ensure access to: • software development environment • technical requirements • 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 program response to specifications • review of candidate design, program and testing documentation • verbal or written questioning to determine candidate's knowledge of: • user requirements • programming techniques • testing techniques.
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>Dynamic data structures</i> may include:	• binary trees • double-linked lists.
<i>Mechanism</i> may include:	• pipes • sockets.
<i>Integrated development environment</i> may include:	• C • Code Warrior • Eclipse • J-Edit • VB • Visual C++ • Visual Studio suite.
<i>Coding standards</i> may include:	• C coding standard • GNU.
<i>Language</i> may include:	• C • C++ • Java • Small Talk • VB • VB.NET.

Unit Sector(s)

Programming and software development