



Australian Government

ICTPRG523 Apply advanced programming skills in another language

Release: 1

ICTPRG523 Apply advanced programming skills in another language

Modification History

Release	Comments
Release 1	This version first released with ICT Information and Communications Technology Training Package Version 1.0.

Application

This unit describes the skills and knowledge required to undertake advanced programming tasks using a selected choice of another programming language. The language may be an object-oriented language.

It applies to individuals who are programmers required to develop program code.

No licensing, legislative or certification requirements apply to this unit at the time of publication.

Unit Sector

Programming and software development

Elements and Performance Criteria

ELEMENT	PERFORMANCE CRITERIA
<i>Elements describe the essential outcomes.</i>	<i>Performance criteria describe the performance needed to demonstrate achievement of the element.</i>
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

ELEMENT	PERFORMANCE CRITERIA
3. Write a simple multi-process application	3.1 Use features of the language that enable inter-process communication through at least one mechanism 3.2 Use 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 the 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 Use source-code version control 6.2 Adhere to guidelines for developing maintainable code, and to a set of provided coding standards 6.3 Apply suitable internal documentation to all the code created, using the tools available in the target language
7. Test the code	7.1 Design and document the tests to be undertaken 7.2 Undertake limited testing of the produced code, to ensure that it complies with the program specification 7.3 Capture the test results
8. Create an application	8.1 Develop a solution, according to program-specification design document 8.2 Design the algorithm, and construct, and test, the application in response to a problem description and language

Foundation Skills

This section describes language, literacy, numeracy and employment skills incorporated in the performance criteria that are required for competent performance.

Skill	Performance Criteria	Description
Reading	4.2, 6.2, 7.2, 8.1	Interprets, and critically analyses, complex texts and

		applies the appropriate strategies to construct meaning from complex texts
Writing	2.2, 6.1, 6.2, 6.3, 7.1	Communicates complex relationships between ideas and information, matching the style of writing to the purpose and audience, and displays a knowledge of structure and layout, employing a broad vocabulary, grammatical structure, and the conventions appropriate to text
Get the work done	All	- Accepts responsibility for planning and sequencing complex tasks and workload, negotiating key aspects with others, taking into account capabilities, efficiencies and effectiveness - Uses systematic, analytical processes in complex, non-routine situations, setting goals, gathering relevant information, and identifying, and evaluating, options against the agreed criteria - Uses analytical processes to decide on a course of action, establishing the criteria for deciding between options - Understands the key principles and concepts, underpinning the design and operation of digital systems and tools

Unit Mapping Information

Code and title current version	Code and title previous version	Comments	Equivalence status
ICTPRG523 Apply advanced programming skills in another language	ICAPRG523A Apply advanced programming skills in another language	Updated to meet Standards for Training Packages	Equivalent unit

Links

Companion Volume implementation guides are found in VETNet -

<https://vetnet.gov.au/Pages/TrainingDocs.aspx?q=a53af4e4-b400-484e-b778-71c9e9d6aff2>

Assessment Requirements for ICTPRG523 Apply advanced programming skills in another language

Modification History

Release	Comments
Release 1	This version first released with ICT Information and Communications Technology Training Package Version 1.0.

Performance Evidence

Evidence of the ability to:

- design and write code using dynamic data structures
- write code using hashing techniques, sorting algorithms and advanced searching techniques
- use language features that enable inter-process communication and responses to operating system signals
- use third-party libraries in the construction of an application, including referencing third-party documentation
- design and build an application program according to a design document
- test and debug code and resolve errors of a syntactical, logical and design origin
- produce documentation according to guidelines and standards.

Note: If a specific volume or frequency is not stated, then evidence must be provided at least once.

Knowledge Evidence

To complete the unit requirements safely and effectively, the individual must:

- identify and describe a range of programming languages
- identify and describe languages for a GUI environment
- summarise the techniques for developing large-scale applications
- outline the procedures for programming with complex data structures
- describe the procedures for using a third-party supplied library, for standard programming features.

Assessment Conditions

Gather evidence to demonstrate consistent performance in conditions that are safe and replicate the workplace. Noise levels, production flow, interruptions and time variances must be typical of those experienced in the programming and software development industry, and include access to:

- the software development environment
- technical documentation.

Assessors must satisfy NVR/AQTF assessor requirements.

Links

Companion Volume implementation guides are found in VETNet -

<https://vetnet.gov.au/Pages/TrainingDocs.aspx?q=a53af4e4-b400-484e-b778-71c9e9d6aff2>