



**Australian Government**

**Department of Education, Employment and Workplace Relations**

# **ICAB5227B Apply advanced programming skills in another language**

**Release: 1**

## ICAB5227B Apply advanced programming skills in another language

### Modification History

Not Applicable

### Unit Descriptor

<b>Unit descriptor</b>	<p>This unit defines the competency required to undertake advanced programming tasks using a selected choice of another programming language. The second language may be an object-oriented language.</p> <p>No licensing, legislative, regulatory or certification requirements apply to this unit at the time of publication.</p>
------------------------	---

### Application of the Unit

<b>Application of the unit</b>	
--------------------------------	--

### Licensing/Regulatory Information

Refer to Unit Descriptor

### Pre-Requisites

<b>Prerequisite units</b>		
	ICAB4229B	Apply intermediate programming skills in another language

## Employability Skills Information

<b>Employability skills</b>	This unit contains employability skills.
-----------------------------	--

## Elements and Performance Criteria Pre-Content

Elements describe the essential outcomes of a unit of competency.	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.
---	--

## Elements and Performance Criteria

ELEMENT	PERFORMANCE CRITERIA
1. Code using advanced data structures	1.1. Design, 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 <i>mechanism</i> 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 GUI environment
5. Debug code	5.1. Use an <i>integrated development environment</i> debugging facilities or a standalone 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 <i>coding standards</i> 6.3. Apply internal documentation suitable for consumption by peers to all code created and utilising documentation tools available in the target <i>language</i>
7. Test code	7.1. Design and document tests to be undertaken 7.2. Undertake limited testing of produced code to ensure program specification is complied with 7.3. Capture test results
8. Create an application	8.1. Develop a solution from a program specification

ELEMENT	PERFORMANCE CRITERIA
	design document 8.2.Design the algorithm, construct and test applications in response to a problem description and <i>language</i>

## Required Skills and Knowledge

### REQUIRED SKILLS AND KNOWLEDGE

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

#### Required skills

- Reading and interpreting program specifications
- Development environment usage
- Working with version control
- Programming techniques
- Documentation techniques
- Advanced debugging techniques
- Advanced testing techniques

#### Required knowledge

- Selected programming language
- Large size application development
- Developing in a GUI environment
- Complex data structures
- Use of third-party supplied library

## Evidence Guide

<b>EVIDENCE GUIDE</b>	
<p>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.</p>	
<b>Overview of assessment</b>	
<b>Critical aspects for assessment and evidence required to demonstrate competency in this unit</b>	<p>Evidence of the following is essential:</p> <ul style="list-style-type: none"> <li>• Assessment must confirm that application programs are designed and built from a provided problem scenario and program specification.</li> <li>• Design and code documentation must be generated. Testing must confirm that created application meets original specification and solves original problem.</li> <li>• Evidence for this competency includes tool usage, documentation, debugging and testing techniques in support of the programming activities.</li> <li>• Assessment must confirm competency in all areas of the software development cycle. Code-only solutions are not acceptable.</li> </ul> <p>To demonstrate competency in this unit the person will require access to:</p> <ul style="list-style-type: none"> <li>• Software development environment</li> <li>• Technical requirements</li> </ul>
<b>Context of and specific resources for assessment</b>	<p>Programming languages form the underpinning software infrastructure of all computer data processing.</p> <p>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.</p> <p>The breadth, depth and complexity covering planning and initiation of alternative approaches to skills or knowledge applications across a broad range of technical and/or management requirements, evaluation and coordination would be characteristic.</p>

<b>EVIDENCE GUIDE</b>	
	<p>Assessment must ensure:</p> <ul style="list-style-type: none"> <li>• The demonstration of competency may also require self-directed application of knowledge and skills, with substantial depth in some areas where judgement is required in planning and selecting appropriate equipment, services and techniques for self and others.</li> <li>• Applications involve participation in development of strategic initiatives as well as personal responsibility and autonomy in performing complex technical operations or organising others. It may include participation in teams including teams concerned with planning and evaluation functions. Group or team coordination may also be involved.</li> </ul>
<b>Method of assessment</b>	<p>The purpose of this unit is to define the standard of performance to be achieved in the workplace. In undertaking training and assessment activities related to this unit, consideration should be given to the implementation of appropriate diversity and accessibility practices in order to accommodate people who may have special needs. Additional guidance on these and related matters is provided in ICA05 Section 1.</p> <ul style="list-style-type: none"> <li>• Competency in this unit should be assessed using summative assessment to ensure consistency of performance in a range of contexts. This unit can be assessed either in the workplace or in a simulated environment. However, simulated activities must closely reflect the workplace to enable full demonstration of competency.</li> <li>• Assessment will usually include observation of real or simulated work processes and procedures and/or performance in a project context as well as questioning on underpinning knowledge and skills. The questioning of team members, supervisors, subordinates, peers and clients where appropriate may provide valuable input to the assessment process. The interdependence of units for assessment purposes may vary with the particular project or scenario.</li> </ul>

<b>EVIDENCE GUIDE</b>	
<b>Guidance information for assessment</b>	<p>Holistic assessment with other units relevant to the industry sector, workplace and job role is recommended.</p> <p>An individual demonstrating this competency would be able to:</p> <ul style="list-style-type: none"> <li>• Demonstrate understanding of a broad knowledge base incorporating theoretical concepts, with substantial depth in some areas</li> <li>• Analyse and plan approaches to technical problems or management requirements</li> <li>• Transfer and apply theoretical concepts and/or technical or creative skills to a range of situations</li> <li>• Evaluate information, using it to forecast for planning or research purposes</li> <li>• Take responsibility for own outputs in relation to broad quantity and quality parameters</li> <li>• Take some responsibility for the achievement of group outcomes</li> <li>• Maintain knowledge of industry products and services</li> </ul> <p>Additionally, an individual demonstrating this competency would be able to:</p> <ul style="list-style-type: none"> <li>• Understand development methodologies and their application in a broad context of programming</li> <li>• Demonstrate advanced theoretical knowledge of language development models</li> <li>• Support development methodology related to a programming project</li> <li>• Thoroughly document all aspects of the programming methodology.</li> </ul>

## Range Statement

### 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



<b>RANGE STATEMENT</b>	
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>Integrated development environment</i> may include but is not limited to:	<ul style="list-style-type: none"> <li>• C</li> <li>• VB</li> <li>• Visual C++</li> <li>• Visual Studio suite</li> <li>• Eclipse</li> <li>• J-Edit</li> <li>• Code Warrior</li> </ul>
<i>Coding standards</i> may include:	<ul style="list-style-type: none"> <li>• C coding standard</li> <li>• GNU coding standard</li> </ul>
<i>Mechanism</i> may include:	<ul style="list-style-type: none"> <li>• sockets</li> <li>• pipes</li> </ul>
<i>Language</i> may include but is not limited to:	<ul style="list-style-type: none"> <li>• C</li> <li>• VB</li> <li>• Java</li> <li>• C++</li> <li>• Small Talk</li> <li>• VB.net</li> </ul>

### Unit Sector(s)

<b>Unit sector</b>	Build
--------------------	-------

### Co-requisite units

<b>Co-requisite units</b>	

## Competency field

Competency field	
------------------	--