



**Australian Government**

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

# **ICAB4222B Apply introductory programming skills in another language**

**Release: 1**

## ICAB4222B Apply introductory programming skills in another language

### Modification History

Not Applicable

### Unit Descriptor

<b>Unit descriptor</b>	<p>This unit defines the competency required to undertake introductory programming tasks using a procedural approach to programming. An object-oriented language may be used in this approach.</p> <p>There may be benefit in concurrent learning with the following unit:</p> <ul style="list-style-type: none"> <li>• ICAB4224B Apply mathematical techniques for software development</li> </ul> <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>		
	ICAB4225B	Automate processes

## 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. Apply basic language syntax and layout	1.1. Demonstrate understanding and application of basic <i>language</i> syntax rules 1.2. Use <i>language</i> data types, operators and expressions to create clear and concise code 1.3. Select and use the appropriate <i>language</i> syntax for sequence, selection and iteration constructs
2. Code using standard algorithms	2.1. Develop algorithms that use the <i>basic programming constructs</i> 2.2. Use a modular programming approach limited to pass-by-value parameters and module return values 2.3. Demonstrate ability to create sequential search, insertion and deletion algorithms to operate on one-dimensional arrays 2.4. Use text files and develop and code standard sequential access algorithms, including end-of-file detection loops
3. Debug code	3.1. Apply standalone debugging tools or tools provided by an <i>integrated development environment</i> to debug code 3.2. Use a debugger to trace code execution and examine variable contents to detect and correct errors
4. Document activities	4.1. Follow <i>organisational guidelines</i> for developing maintainable code and adhere to the provided <i>coding standard</i> when documenting activities 4.2. Apply internal documentation suitable for consumption by peers to code created and utilise documentation tools available in the target <i>language</i> when documenting activities
5. Test code	5.1. Create and conduct simple tests to confirm code meets design specification 5.2. Document the tests performed and results achieved
6. Create an application	6.1. Develop a solution when provided with a basic design document, including a program specification 6.2. Design the algorithm, construct and test applications in response to a problem description and language elements

## 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
- Translating requirements from problem space to machine space
- Integrated development environment usage
- Basic programming techniques
- Internal (code) documentation techniques
- Basic debugging techniques
- Testing techniques
- Basic documentation techniques

#### Required knowledge

- Programming language
- Small size application development
- Data structures
- GUI interfaces
- Best practice in application of language syntax rules

## Evidence Guide

<b>EVIDENCE GUIDE</b>	
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.	
<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 be used to confirm that created application meets original specifications and solves original problem.</li> <li>• Assessment must confirm competency in all areas of the software development cycle. Code only solutions are not acceptable.</li> <li>• Questions related to the performance criteria will assist in assessing competency. Observation of skills may assist in the collection of evidence.</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>The breadth, depth and complexity of knowledge and skills in this competency would cover a broad range of varied activities or application in a wider variety of contexts most of which are complex and non-routine. Leadership and guidance would be involved when organising activities of self and others as well as contributing to technical solutions of a non-routine or contingency nature.</p> <p>Assessment must ensure:</p> <ul style="list-style-type: none"> <li>• Performance of a broad range of skilled applications</li> </ul>

<b>EVIDENCE GUIDE</b>	
	<p>including the requirement to evaluate and analyse current practices, develop new criteria and procedures for performing current practices and provision of some leadership and guidance to others in the application and planning of the skills would be characteristic.</p> <ul style="list-style-type: none"> <li>• Applications may involve responsibility for, and limited organisation of, others</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>Guidance information for assessment</b>	<p>Holistic assessment with other units relevant to the industry sector, workplace and job role is recommended, such as:</p> <ul style="list-style-type: none"> <li>• ICAB4224B Apply mathematical techniques for software development</li> </ul>

<b>EVIDENCE GUIDE</b>	
	<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 some theoretical concepts</li> <li>• Apply solutions to a defined range of unpredictable problems</li> <li>• Identify and apply skill and knowledge areas to a wide variety of contexts, with depth in some areas</li> <li>• Identify, analyse and evaluate information from a variety of sources</li> <li>• Take responsibility for own outputs in relation to specified quality standards</li> <li>• Take limited responsibility for the quantity and quality of the output of others</li> <li>• Maintain knowledge of industry products and services</li> </ul>

## Range Statement

<b>RANGE STATEMENT</b>	
<p>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.</p>	
<p><b><i>Organisational guidelines</i></b> may include but are not limited to:</p>	<ul style="list-style-type: none"> <li>• personal use of emails and internet access</li> <li>• content of emails</li> <li>• downloading information and accessing particular websites</li> <li>• opening mail with attachments</li> <li>• virus risk</li> <li>• dispute resolution</li> <li>• document procedures and templates</li> <li>• communication methods</li> <li>• financial control mechanisms</li> </ul>
<p><b><i>Basic programming constructs</i></b></p>	<ul style="list-style-type: none"> <li>• validation loops</li> </ul>



<b>RANGE STATEMENT</b>	
may include but are not limited to:	<ul style="list-style-type: none"> <li>• sentinel-controlled loops</li> <li>• nested selections</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>
<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 standard</i> may include:	<ul style="list-style-type: none"> <li>• Java coding standard</li> <li>• GNU coding standard</li> </ul>

### Unit Sector(s)

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

### Co-requisite units

<b>Co-requisite units</b>	

## Competency field

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