



Australian Government

Department of Education, Employment and Workplace Relations

ICAB4219B Apply introductory object-oriented language skills

Release: 1

ICAB4219B Apply introductory object-oriented language skills

Modification History

Not Applicable

Unit Descriptor

Unit descriptor	<p>This unit defines the competency required to undertake introductory programming tasks using an object-oriented programming language. Competency includes tool usage, documentation, debugging and testing techniques in support of the programming activities.</p> <p>There may be benefit in concurrent learning with the following unit:</p> <ul style="list-style-type: none"> • ICAB4224B Apply mathematical techniques for software development <p>The following unit is linked and forms an appropriate cluster:</p> <ul style="list-style-type: none"> • ICAA4058B Apply skills in object oriented design <p>No licensing, legislative, regulatory or certification requirements apply to this unit at the time of publication.</p>
------------------------	---

Application of the Unit

Application of the unit	
--------------------------------	--

Licensing/Regulatory Information

Refer to Unit Descriptor

Pre-Requisites

Prerequisite units		
	ICAB4225B	Automate processes

Employability Skills Information

Employability skills	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 and best practices 1.2. Select and use <i>language</i> data types, operators and expressions to create clear and concise code 1.3. Use the appropriate <i>language</i> syntax for sequence, selection and iteration constructs 1.4. Use a modular programming approach within member/function logic 1.5. Apply arrays, including arrays of objects to introductory programming tasks 1.6. Use <i>standard array processing algorithms</i> 1.7. Use the facilities of the <i>language</i> to read and write data from/to text files and record outcomes
2. Apply basic object-oriented principles in the target language	2.1. Implement a class that contains primitive member/instance variables 2.2. Implement a class that contains multiple options for object construction 2.3. Implement a class that uses user-defined aggregation (object instance/member variables) 2.4. Use the facilities provided in the <i>language</i> to implement inheritance to at least two levels of depth 2.5. Use polymorphism at a simple level through inheritance to enable easy extension of code
3. Debug code	3.1. Use an <i>integrated development environment</i> , in particular the <i>language</i> debugging facilities, to debug code 3.2. Interpret compiler/interpreter messages to resolve syntax errors and use debugging techniques to resolve logic errors
4. Document activities	4.1. Follow organisational guidelines for developing maintainable code and adhere to the provided <i>coding standards</i> when documenting activities 4.2. Apply internal documentation to all 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 <i>design specification</i> 5.2. Document the tests performed and results achieved
6. Create an application	6.1. Develop a solution when provided with a basic

ELEMENT	PERFORMANCE CRITERIA
	<p data-bbox="619 297 1056 333" style="text-align: center;"><i>object-oriented design document</i></p> <p data-bbox="576 342 1206 418" style="text-align: center;">6.2. Reference appropriate documentation for the <i>language</i></p>

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
- Programming techniques
- Internal (code) documentation techniques
- Debugging techniques
- Testing techniques
- Documentation techniques

Required knowledge

- Object-oriented programming concepts
- Object-oriented programming language
- Small size application development
- Using a GUI to interact with operator

Evidence Guide

EVIDENCE GUIDE	
<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>	
Overview of assessment	
Critical aspects for assessment and evidence required to demonstrate competency in this unit	<p>Evidence of the following is essential:</p> <ul style="list-style-type: none"> • Assessment must confirm that application programs are designed and built from a provided problem scenario and program specification. • Code documentation must be generated. Testing must be used to confirm that created application meets original specification and solves original problem. <p>To demonstrate competency in this unit the person will require access to:</p> <ul style="list-style-type: none"> • Software development environment • Technical requirements
Context of and specific resources for assessment	<p>Programming in object-oriented languages is a software development methodology that offers the programmer standard reusable software modules (components), rather than requiring the developer to write custom programming code each time.</p> <p>Using standard components reduces development time (because the writing and testing of those components has already been done by other programmers), and ensures a standard look and feel for programs using the same components. Object-oriented languages are an important feature of software development processes world-wide.</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>

EVIDENCE GUIDE	
	<p>Assessment must ensure:</p> <ul style="list-style-type: none"> • Performance of a broad range of skilled applications 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. • Applications may involve responsibility for, and limited organisation of, others. <p>The stages of the development methodology should be followed within the scope of a project or scenario, and the relevant supporting documentation produced.</p>
Method of assessment	<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"> • 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. • 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

EVIDENCE GUIDE	
	<p>purposes may vary with the particular project or scenario.</p>
Guidance information for assessment	<p>Holistic assessment with other units relevant to the industry sector, workplace and job role is recommended, for example:</p> <ul style="list-style-type: none"> • ICAB4224B Apply mathematical techniques for software development • ICAA4058B Apply skills in object-oriented design <p>An individual demonstrating this competency would be able to:</p> <ul style="list-style-type: none"> • Demonstrate understanding of a broad knowledge base incorporating some theoretical concepts • Apply solutions to a defined range of unpredictable problems • Identify and apply skill and knowledge areas to a wide variety of contexts, with depth in some areas • Identify, analyse and evaluate information from a variety of sources • Take responsibility for own outputs in relation to specified quality standards • Take limited responsibility for the quantity and quality of the output of others • Maintain knowledge of industry products and services <p>Additionally, an individual demonstrating this competency would be able to:</p> <ul style="list-style-type: none"> • Understand a range of development object-oriented methodologies and their application • Demonstrate theoretical knowledge of object-oriented language development • Apply a methodology to a project or scenario • Produce documentation required by the chosen methodology

Range Statement

RANGE STATEMENT	
<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><i>Integrated development environment</i> may include but is not limited to:</p>	<ul style="list-style-type: none"> • Visual C++ • Visual Studio suite • Eclipse • J-Edit • Code Warrior • JBuilder
<p><i>Coding standards</i> may include:</p>	<ul style="list-style-type: none"> • Java coding standard • GNU coding standard
<p><i>Language</i> may include but is not limited to:</p>	<ul style="list-style-type: none"> • Java • C++ • Small Talk • VB.net • C#.net
<p><i>Design specification</i> may include but is not limited to:</p>	<ul style="list-style-type: none"> • technical requirements • user problem statement • current system functionality
<p><i>Standard array processing algorithms</i> may include but is not limited to:</p>	<ul style="list-style-type: none"> • search • insertion and deletion algorithms
<p><i>Object-oriented design document</i> may include:</p>	<ul style="list-style-type: none"> • use-cases • class diagrams • supplementary specifications

Unit Sector(s)

Unit sector	Build
-------------	-------

Co-requisite units

Co-requisite units		

Competency field

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