

## ICAB4219A

## Apply introductory object-oriented language skills

### Unit Descriptor

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.

There may be benefit in concurrent learning with the following units:

ICAB4224A Apply mathematical techniques for software development  
Build

### Unit Sector

### ELEMENT

### PERFORMANCE CRITERIA

- |   |  |
|---|--|
| 1. Apply basic language syntax and layout                       | <ul style="list-style-type: none"> <li>1.1 Demonstrate understanding and application of basic <i>language</i> syntax rules and best practices</li> <li>1.2 Select and use <i>language</i> data types, operators and expressions to create clear and concise code</li> <li>1.3 Use the appropriate <i>language</i> syntax for sequence, selection and iteration constructs</li> <li>1.4 Use a modular programming approach within member/function logic</li> <li>1.5 Apply arrays, including arrays of objects to introductory programming tasks</li> <li>1.6 Use <i>standard array processing algorithms</i></li> <li>1.7 Use the facilities of the <i>language</i> to read and write data from/to text files and record outcomes</li> </ul> |
| 2. Apply basic objectoriented principles in the target language | <ul style="list-style-type: none"> <li>2.1 Implement a class that contains primitive member/instance variables</li> <li>2.2 Implement a class that contains multiple options for object construction</li> <li>2.3 Implement a class that uses user-defined aggregation (object instance/member variables)</li> <li>2.4 Use the facilities provided in the <i>language</i> to implement inheritance to at least two levels of depth</li> <li>2.5 Use polymorphism at a simple level through inheritance to enable easy extension of code</li> </ul>   |
| 3. Debug code   | <ul style="list-style-type: none"> <li>3.1 Use an <i>integrated development environment</i>, in particular the <i>language</i> debugging facilities, to debug code</li> <li>3.2 Interpret compiler/interpreter messages to resolve syntax errors and use debugging techniques to resolve logic errors</li> </ul>   |
| 4. Document activities  | <ul style="list-style-type: none"> <li>4.1 Follow organisational guidelines for developing maintainable code and adhere to the provided <i>coding standards</i> when documenting activities</li> <li>4.2 Apply internal documentation to all code created and utilise documentation tools available in the target <i>language</i> when documenting activities</li> </ul>   |

- |                          |  |
|--------------------------|--|
| 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 <i>object-oriented design document</i> |
|                          | 6.2 Reference appropriate documentation for the <i>language</i>                          |

## KEY COMPETENCIES

The seven Key Competencies represent generic skills considered necessary for effective participation by an individual in the workplace.

Performance Level 1 at this level, the candidate is required to undertake tasks effectively

Performance Level 2 at this level, the candidate is required to manage tasks

Performance Level 3 at this level, the candidate is required to use concepts for evaluating and reshaping tasks

The following Key Competency levels have been considered within the structure of this unit's Performance Criteria.

Key Competency	Performance Level
Communicating ideas and information	2
Collecting analysing and organising information	2
Planning and organising activities	2
Working with others and in teams	1
Using mathematical ideas and techniques	2
Solving problems	3
Using technology	3

## RANGE STATEMENT

The Range Statement contextualises the unit of competency and provides a focus for assessment. The information provided is intended to define the scope of assessment and to assist assessors define the performance to be achieved by an individual in the workplace. The Range Statement relates to the unit of competency as a whole. It allows for different work environments and situations that may affect performance. Italicised wording in the Performance Criteria is detailed as follows.

Integrated development environment	May include but is not limited to Visual C++, Visual Studio suite, Eclipse, J-Edit, Code Warrior, JBuilder
Coding standards	May include Java coding standard, GNU coding standard
Language	May include but is not limited to Java, C++, Small Talk, VB.net, C#.net
Design specification	May include but is not limited to technical requirements, user problem statement, current system functionality

Standard array processing algorithms	May include but are not limited to search, insertion and deletion algorithms
Objectoriented design document	May include use-cases, class diagrams and supplementary specifications

## EVIDENCE GUIDE

The Evidence Guide provides advice on assessment and must be read in conjunction with the Performance Criteria, required skills and knowledge, the Range Statement and the assessment guidelines for this Training Package.

### Critical aspects of evidence

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.

### Knowledge and skills

Knowledge includes:

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

Skills include:

- 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

**Assessment guidance**

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 Introduction.

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 purposes may vary with the particular project or scenario.

In the case of this unit, it could be assessed in a holistic manner with:

- ICAA4058A Apply skills in object-oriented design

**Resources**

To demonstrate competency in this unit the person will require access to:

- Software development environment
- Technical requirements

**Role context**

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. 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. 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.

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.

An individual demonstrating this competency would be able to:

- 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

The stages of the development methodology should be followed within the scope of a project or scenario, and the relevant supporting documentation produced.

Additionally, an individual demonstrating this competency would be able to:

- 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