



Australian Government

Department of Education, Employment and Workplace Relations

ICAPRG501A Apply advanced object-oriented language skills

Release: 1

ICAPRG501A Apply advanced object-oriented language skills

Modification History

Release	Comments
Release 1	This Unit first released with <i>ICAll Information and Communications Technology Training Package version 1.0</i>

Unit Descriptor

This unit describes the performance outcomes, skills and knowledge required to undertake advanced programming tasks using an object-oriented programming language.

Application of the Unit

This unit applies to programmers who are required to produce complex object-oriented programming.

Licensing/Regulatory Information

No licensing, legislative, regulatory or certification requirements apply to this unit at the time of endorsement but users should confirm requirements with the relevant federal, state or territory authority.

Pre-Requisites

Not applicable.

Employability Skills Information

This unit contains employability skills.

Elements and Performance Criteria Pre-Content

Element	Performance Criteria
<i>Elements describe the essential outcomes of a unit of competency.</i>	<i>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.</i>

Elements and Performance Criteria

1. Develop client-server application	<p>1.1 Build and test object-oriented applications</p> <p>1.2 Make use of the features of the language that enable inter-process communication through at least one <i>mechanism</i></p> <p>1.3 Make use of the features of the language that enable remote procedure calls (RPC) using a class that is based on multiple inheritances</p> <p>1.4 Build programs using classes that are based on nested classes</p>
2. Develop graphical user interface (GUI)	<p>2.1 Code to implement drag and drop</p> <p>2.2 Develop GUI help files</p> <p>2.3 Code using 2-D graphics</p>
3. Build applications	<p>3.1 Code within an existing <i>architectural framework</i></p> <p>3.2 Ensure code follows standards for object-oriented language</p> <p>3.3 Develop an application that includes data transfer between client and server</p> <p>3.4 Review the concept of design patterns used by the architectural framework</p>
4. Debug code	<p>4.1 Use stand-alone debugging tools or tools provided by <i>integrated development environment</i> to examine running code</p> <p>4.2 Detect logical and coding errors using debugger</p> <p>4.3 Detect and correct errors by tracing code and examining variable content</p>
5. Test application	<p>5.1 Design and document tests to be undertaken</p> <p>5.2 Undertake limited testing of produced code to ensure it complies with program specification</p> <p>5.3 Document test results</p>
6. Document system	<p>6.1 Demonstrate adherence to guidelines for developing maintainable code and company or institutional <i>coding standards</i></p> <p>6.2 Create code using supplied design documents</p> <p>6.3 Create and maintain program documentation</p> <p>6.4 Ensure that user documentation in the form of online help is built into applications</p>

Required Skills and Knowledge

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

Required skills

- analytical skills to interpret design specifications, translating requirements from problem space to machine space
- literacy skills to interpret and write technical documents
- planning and organisational skills to produce work within an agreed schedule
- problem-solving skills to debug code
- technical skills to:
 - follow coding standards
 - produce client-server application development
 - produce internal (code) documentation techniques
 - use an integrated development environment (IDE)
 - use good programming techniques
 - use web-enabled application development
 - use debugging techniques
 - write a GUI to interact effectively with operator.

Required knowledge

- detailed knowledge of:
 - architecture of a framework for web-enabled application development
 - techniques for implementing inter-process communication
 - large-size application development
 - testing techniques as applied to distributed application development
 - techniques for implementing third-party supplied code.

Evidence Guide

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.

Overview of assessment	
Critical aspects for assessment and evidence required to demonstrate competency in this unit	<p>Evidence of the ability to:</p> <ul style="list-style-type: none"> • design an application to meet user requirements using object-oriented techniques • build an application program using a range of tools for a given problem • test application to ensure that it meets client requirements • document the application.
Context of and specific resources for assessment	<p>Assessment must ensure access to:</p> <ul style="list-style-type: none"> • database management system (DBMS) • networked computers • software-development environment • technical requirements • appropriate learning and assessment support when required • modified equipment for people with special needs.
Method of assessment	<p>A range of assessment methods should be used to assess practical skills and knowledge. The following examples are appropriate for this unit:</p> <ul style="list-style-type: none"> • evaluation of completed application • review of: <ul style="list-style-type: none"> • application documentation • testing documentation • verbal or written questioning to determine knowledge of: <ul style="list-style-type: none"> • object-oriented techniques and terms • DBMS access • software testing.
Guidance information for assessment	<p>Holistic assessment with other units relevant to the industry sector, workplace and job role is recommended, where appropriate.</p> <p>Assessment processes and techniques must be culturally appropriate, and suitable to the communication skill level, language, literacy and numeracy capacity of the candidate and the work being performed.</p> <p>Indigenous people and other people from a non-English speaking</p>

	background may need additional support. In cases where practical assessment is used it should be combined with targeted questioning to assess required knowledge.
--	--

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 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>Mechanism</i> may include:	<ul style="list-style-type: none"> • pipes • sockets.
<i>Architectural framework</i> may include:	<ul style="list-style-type: none"> • J2EE • .NET.
<i>Integrated development environment</i> may include:	<ul style="list-style-type: none"> • Code Warrior • Eclipse • JBuilder • J-Edit • Visual C++ • Visual Studio suite • WebSphere.
<i>Coding standards</i> may include:	<ul style="list-style-type: none"> • GNU • Java.

Unit Sector(s)

Programming and software development