Australian Government

Department of Education, Employment and Workplace Relations

ICAPRG527A Apply intermediate object-oriented language skills

Release: 1

ICAPRG527A Apply intermediate object-oriented language skills

Modification History

Release	Comments
Release 1	This Unit first released with *ICA11 Information and Communications Technology Training Package version 1.0*

Unit Descriptor

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

Application of the Unit

This unit applies to programmers in a variety of fields who are required to produce programs in object-oriented languages.

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
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

1. Build applications using provided language utilities	1.1 Review program requirements 1.2 Divide multiple source-code files into logical units and packages 1.3 Use at least two of the utilities of the target language allowing for internal storage of ***collections of data*** 1.4 Use the utilities of the target language providing internal data sorting and searching facilities 1.5 Employ ***integrated-development environment*** facilities to make files to automate program building 1.6 Follow guidelines for developing maintainable code adhering to ***coding standards*** 1.7 Use the facilities in the language for persisting objects to binary files 1.8 Use the operator and function or method overloading facilities available in the ***language*** at an introductory level 1.9 Demonstrate ability to use exception-handling techniques to ensure program stability 1.10 Demonstrate use of a class that is based on multiple inheritances
2. Write programs that interact with a database	2.1 Design and implement programs that connect to a ***database*** 2.2 Design and implement programs that use the language facilities to extract, update and delete data stored in a database 2.3 Design and implement programs that use the language facilities to manipulate database structure (query, create and delete) 2.4 Write programs that deliver transactional integrity
3. Write graphical user interface (GUI)	3.1 Employ GUI framework or text windowing interface appropriate to the chosen language 3.2 Demonstrate use of standard ***GUI components*** 3.3 Use the facilities within the language for GUI objects to respond to user and program-generated events
4. Debug application	4.1 Use stand-alone debugging tools or tools provided by integrated development environment to examine variables and trace running code 4.2 Use debugger to detect logical and coding errors 4.3 Use tracing of code and examination of variable contents

	during execution to detect and correct errors
5. Test application	5.1 Design and document limited tests of code 5.2 Undertake limited testing of produced code to ensure it complies with program specification 5.3 Capture and document test results
6. Create and maintain documentation	6.1 Read and interpret supplied design document to create code 6.2 Create and maintain program documentation

Required Skills and Knowledge

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

Required skills

- analytical skills to analyse requirements document
- communication skills to communicate with user and colleagues
- initiative and enterprise skills to produce an appropriate application to fulfil requirements
- literacy skills to:
  - produce program documentation
  - read requirements documentation
- planning and organisational skills to produce output in a timely fashion as required
- problem-solving skills to debug program code
- technical skills to:
  - interpret program specifications
  - translate requirements from problem space to machine space
  - integrate development environment usage
  - use internal documentation techniques
  - test program to ensure it meets its requirements.

Required knowledge

- detailed knowledge of:
  - data structures
  - small-size and medium-size application development
  - object-oriented programming concepts
  - object-oriented programming language
  - process and techniques related to the use of GUI to interact with operator
- documentation techniques to document the application.

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	Evidence of the ability to: • design and build application programs from a problem scenario and program specification.
Context of and specific resources for assessment	Assessment must ensure access to: • requirements document or object-oriented design structures • programming languages that support object-oriented development • database management system (DBMS) • appropriate learning and assessment support when required • modified equipment for people with special needs.

Method of assessment	A range of assessment methods should be used to assess practical skills and knowledge. The following examples are appropriate for this unit:
	• evaluation of completed application
	• review of documentation of this application
	• verbal or written questioning to ensure knowledge of:
	  • object-oriented programming
	  • coding standards
	  • methods used to access data in databases
	  • documentation techniques.

Guidance information for assessment	Holistic assessment with other units relevant to the industry sector, workplace and job role is recommended, where appropriate. 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. Indigenous people and other people from a non-English speaking 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.

Collections of data may include:	• hash tables • lists • queues • sets • stacks • trees.
Integrated development environment may include:	• Code Warrior • Eclipse • J-Edit • Visual C++ • Visual Studio suite.
Coding standards may include:	• GNU • Java.
Language may include:	• C#.net • C++ • Java • Small Talk • VB.NET.
Database may be:	• object • relational.
GUI components may include:	• buttons • check boxes • drop-down lists • option buttons • text input fields.
Documentation may include:	• architecture documentation • code comments • design documents • in-code documentation • internal module documentation • release documents • requirement documents • test documents

	• user manuals.

Unit Sector(s)

Programming and software development