Australian Government

ICTPRG406 Apply introductory object-oriented language skills

Release: 1

INDUSTRY SKILLS COUNCILS
Creating Australia's Future

ICTPRG406 Apply introductory object-oriented language skills

Modification History

Release	Comments
Release 1	This version first released with ICT Information and Communications Technology Training Package Version 1.0.

Application

This unit describes the performance outcomes, skills and knowledge required to undertake introductory programming tasks using an object-oriented programming language, including tool usage, documentation, debugging, and testing techniques.

It applies to individuals who are programmers in a variety of fields and who are required to produce simple programs in object-oriented languages.

No licensing, legislative or certification requirements apply to this unit at the time of publication.

Unit Sector

Programming and software development

Elements and Performance Criteria

ELEMENT	PERFORMANCE CRITERIA
Elements describe the essential outcomes.	*Performance criteria describe the performance needed to demonstrate achievement of the element.*
1. Apply basic language syntax and layout	1.1 Apply basic language syntax rules and best practices 1.2 Select and use language data types, operators and expressions, in order to create clear and concise code 1.3 Use the appropriate language syntax for sequence, selection and iteration constructs 1.4 Use a modular programming approach within member or function logic 1.5 Apply arrays, including arrays of objects to introductory programming tasks 1.6 Use standard-array processing algorithms 1.7 Use the facilities of the language to read and write data, from and to, text files, and record the outcomes
2. Apply basic object-oriented principles in the target language	2.1 Implement a class that contains primitive member or 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 or member variables) 2.4 Use the facilities of the language to implement inheritance, to at least two levels 2.5 Use polymorphism at a simple level through inheritance, to enable the easy extension of the code
3. Debug the code	3.1 Use the language debugging facilities of an integrated development environment (IDE) 3.2 Interpret the compiler or interpreter messages to resolve syntax errors, and use debugging techniques to resolve logic errors
4. Document the activities	4.1 Follow organisational guidelines for developing maintainable code, and adhere to the provided coding standards, when documenting activities 4.2 Apply internal documentation to all the code created, and use the documentation tools available in the target language, when documenting activities

ELEMENT	PERFORMANCE CRITERIA
5. Test the code	5.1 Create and conduct simple tests, to confirm that the code meets the design specification 5.2 Document the tests performed and the results achieved
6. Create an application	6.1 Develop a solution, when provided with a basic object-oriented design document 6.2 Refer to the appropriate documentation for the language

Foundation Skills

This section describes language, literacy, numeracy and employment skills incorporated in the performance criteria that are required for competent performance.

Skill	**Performance Criteria**	**Description**
Reading	3.2, 4.1, 5.1, 6.2	• Evaluates, and integrates, information and ideas to construct meaning and selects, and applies, a range of reading strategies in relation to design specifications, coding standards, and coding- language documentation
Writing	1.1-1.7, 4.1, 4.2, 5.2	• Communicates relationships between ideas and information, in a style appropriate to the audience and purpose, and selects the vocabulary, grammatical structures and conventions appropriate to the text, in relation to coding, recording outcomes, and documenting activities
Numeracy	1.2, 1.3, 1.4, 1.5, 1.6, 2.1, 2.2, 2.3, 2.4, 2.5	• Selects from, and flexibly applies, mathematical and problem-solving strategies and techniques, in a programming context • Uses formal written mathematical language and representation, in the context of programming
Navigate the world of work	1.1, 4.1	• Recognises and follows, explicit and implicit protocols, and meets expectations associated with own role, when developing code that is compliant with standards and organisational guidelines
Get the work done	1.1, 1.2, 3.1, 3.2, 4.2, 5.1, 6.1	• Uses a formal decision-making process, identifying and evaluating several choices against a limited set of criteria, when selecting language data types, operators and expressions • Evaluates the effectiveness of decisions, in terms of how well they meet the stated design specifications • Uses analytical processes to decide on a course of action, when translating requirements from the problem space to machine space, and when debugging • Utilises a broad range of features within applications in order to develop software programs • Recognises, and uses, a wide range of relevant language and symbols, when applying the coding syntax • Actively identifies systems, devices and applications with the potential to meet current and/or future needs regarding programming

Unit Mapping Information

Code and title **current version**	**Code and title** **previous version**	**Comments**	**Equivalence status**
ICTPRG406 Apply introductory object-oriented language skills	ICAPRG406A Apply introductory object-oriented language skills	Updated to meet Standards for Training Packages	Equivalent unit

Links

Companion volumes available from the IBSA website: http://www.ibsa.org.au/companion_volumes - https://vetnet.education.gov.au/Pages/TrainingDocs.aspx?q=a53af4e4-b400-484e-b778-71c9e9d6aff2