



Australian Government

Department of Education, Employment and Workplace Relations

ICAPRG526A Maintain functionality of legacy code programs

Release: 1

ICAPRG526A Maintain functionality of legacy code programs

Modification History

Release	Comments
Release 1	This Unit first released with <i>ICA11 Information and Communications Technology Training Package version 1.0</i>

Unit Descriptor

This unit describes the performance outcomes, skills and knowledge required to maintain the functionality of legacy code programs.

Application of the Unit

This unit applies to individuals working in software development who are required to maintain legacy programs, updating them as required.

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. Determine functional changes required	1.1 Consult user to determine and document the functional change required 1.2 Clarify impact of changes with operational personnel
2. Review legacy language	2.1 Identify the basic structures of the legacy language 2.2 Review the data storage of the legacy language with regard to global variables, modularisation, cohesion and coupling 2.3 Review the basic algorithms of the legacy language
3. Identify and isolate change	3.1 Review the technical documentation for the system to determine modules to change 3.2 Review source code to determine program logic and flow 3.3 Update the technical documentation to reflect current state of program
4. Determine framework for change	4.1 Determine the change approach to be used to implement the change 4.2 Ascertain and detail the risks for each approach 4.3 Determine the personnel who will make the changes
5. Design change	5.1 Develop the changes required, considering the impact of the changes on other parts of the system 5.2 Document the changes according to change-management standards
6. Code change	6.1 Identify and obtain appropriate hardware and software for the change 6.2 Write code according to programming standards 6.3 Compile and unit test the changes 6.4 Document the changes according to programming standards
7. Test change	7.1 System test the changes and record outcomes 7.2 Perform user acceptance testing and record outcomes 7.3 Sign off change request
8. Archive system	8.1 Create new version of software according to configuration management standards 8.2 Archive the development system, including source code, compilers and test data

Required Skills and Knowledge

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

Required skills

- analytical skills to determine functional changes and impacts these changes will have on the system
- communication skills to liaise with user and operational personnel
- literacy skills to interpret and write technical documentation
- planning and organisational skills to determine change-management process
- technical skills to:
 - code in legacy language
 - test new code
 - use change-management system.

Required knowledge

- operational procedures on the required operating system
- organisational standards for software development
- security and viruses
- privacy legislation and copyright with regard to legacy code, the particular system involved and its data
- technical terminology related to reading help files and manuals
- types of software.

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: • read legacy language • identify and isolate the required change • implement the change via the chosen technique.
Context of and specific resources for assessment	Assessment must ensure access to: • documents detailing organisational standards for software development • existing legacy code • appropriate compilers • 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 candidate's completed change in legacy code • review of candidate's documentation that reflects modification in legacy programs • verbal or written questioning to determine candidate's knowledge of: • key design alternatives • change-management process.
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.

User may include:	• department within the organisation • person within a department • third party.
Technical documentation may include:	• brochures • help references • online help • project specifications • reports • technical manuals • training materials and self-paced tutorials • user guides.
Change approach may include:	• implementing a new layer or replacing an existing layer to the architecture of the system • implementing the changes in the legacy language • implementing wrappers to the legacy code.
Hardware may include:	• modems and other connectivity devices, such as asymmetric digital subscriber line (ADSL) modems • networks • personal computers • remote sites • servers • workstations.
Software may include:	• commercial applications • customised • in-house • organisation-specific • packaged.

Unit Sector(s)

Programming and software development