

Unit Title:	Software design fundamentals
OCR unit number:	33
Unit reference number:	L/601/3251
Level:	3
Credit value:	10
Guided learning hours:	80

Unit aim

The aim of this unit is that learners will:

- Understand the principles of software design
- Apply the techniques of software design

Learning Outcomes	Assessment Criteria	Knowledge, understanding and skills
<p>The Learner will:</p> <p>1 Understand the principles of software design</p>	<p>The Learner can:</p> <p>1.1 Describe the role of software design and computer programming in the IT Systems Development Life Cycle (SDLC)</p> <p>1.2 Describe the application and limits of programming paradigms procedural, object oriented and event driven and the available supporting tools and environments (e.g. CASE tools, IDEs)</p> <p>1.3 Explain sequence, selection and iteration as used in computer programming</p> <p>1.4 Explain abstraction of data and code and the use of predefined data and code in computer programming</p> <p>1.5 Explain the importance of the readability and understandability of code and how these can be improved by naming, comments and layout</p>	<ul style="list-style-type: none"> • the differing stages of the SDLC and in which stage(s) software design and computer programming are used and the role of these within each stage • the range of programming paradigms that can be used during the development of software. The paradigms should include: <ul style="list-style-type: none"> - procedural - object oriented - event driven - the available supporting tools and environments • the concepts of sequence, selection and iteration when used during at least 3 different software programming • how these concepts are used to achieve efficient programming • the development, analysing and testing of data and code • the use of predefined code and data

Learning Outcomes	Assessment Criteria	Knowledge, understanding and skills
	<p>1.6 Describe how the following factors contribute to the quality of code: efficiency, reliability, robustness, usability, portability and maintainability</p>	<ul style="list-style-type: none"> • the use of libraries of code and data • the advantages and disadvantages of using libraries of code and data • how readability and understandability of code can be enhanced by the use of naming conventions, comments and the layout of the code • the importance of using these concepts when developing code to enable the code to be easily read and understood by a third party • the differing factors that can contribute to the quality of code being produced. These factors should include: <ul style="list-style-type: none"> - efficiency - reliability - robustness - usability - portability - maintainability • how these factors can have a detrimental effect on the quality of code
<p>2 Apply the techniques of software design</p>	<p>2.1 Develop algorithms to represent problems</p> <p>2.2 Identify and define data and file storage requirements including predefined data items</p> <p>2.3 Identify and define program structures including predefined code items</p> <p>2.4 Identify and represent required inputs and outputs</p> <p>2.5 Use tools (e.g. pseudocode) to express software designs</p>	<ul style="list-style-type: none"> • the range of algorithms and their use during software design • how to develop algorithms to represent problems including: <ul style="list-style-type: none"> - high level descriptive - recursive/iterative - logical • how to identify and define data and file storage requirements • the types (e.g. string, Boolean) and length of data should be defined prior to storage requirements being defined

Learning Outcomes	Assessment Criteria	Knowledge, understanding and skills
		<ul style="list-style-type: none"> • how to identify and define a range of program structures to include: <ul style="list-style-type: none"> - modular - procedural - event driven • the differing structures which are available when using predefined code items. The candidates should understand, and be able to identify and define, the correct program structures when using predefined code • how to use tools to express software designs. Be aware of the range of tools that are available, for example: <ul style="list-style-type: none"> - pseudocode - JSP diagrams - structure diagrams • there are many different tools that can be used to express software designs. Candidates should select the most appropriate for the software

Assessment

The qualification has been designed to develop knowledge, understanding and skills in the full range of functions involved in the planning and control, hardware, software and systems installation, software solutions and the production of customer support materials. It also provides opportunities for learners to study towards system and network management, to specialise in one or more specific programming languages in addition to being able to take units that are vendor specific.

Each unit within the specification is designed around the principle that candidates will build a portfolio of evidence relating to progression towards meeting the unit assessment criteria.

The unit assessment criteria reflect the demands of the learning outcomes for each unit.

In order for candidates to be able to effectively progress towards meeting the requirements of each assessment criteria, tutors must make sure that the supporting knowledge, understanding and skills requirements for each criteria are fully addressed. The identified knowledge, understanding and skills are not exhaustive and may be expanded upon or tailored to particular contexts to which the unit is being taught and the assessment criteria applied.

We recommend that teaching and development of subject content and associated skills be referenced to real vocational situations, through the utilisation of appropriate industrial contact, vocationally experienced delivery personnel, and real life case studies.

All the learning outcomes and assessment criteria must be clearly evidenced in the submitted work, which is externally moderated by OCR.

Results will be Pass or Fail.

Guidance on assessment

Candidates do not have to achieve units in any particular order and tutors should tailor learning programmes to meet individual candidate needs. It is recommended that, wherever possible, centres adopt a holistic approach to the delivery of the qualification and identify opportunities to link the units.

Centres are free to deliver this qualification using any mode of delivery that meets the needs of their candidates. Whatever mode of delivery is used, centres must ensure that learners have access to appropriate resources and consider the candidates' complete learning experience when designing learning programmes. This is particularly important in relation to candidates studying part time alongside real work commitments where candidates may bring with them a wealth of experience that should be utilised to maximum effect by tutors and assessors.

It is difficult to give a detailed answer to how much evidence is required as it depends on the type of evidence collected and the judgement of assessors. The main principles, however, are as follows: for a candidate to be judged competent in a unit, the evidence presented must satisfy:

- all the items listed, in the section 'Learning Outcomes'
- all the areas in the section 'Assessment Criteria'

Questioning the candidate is normally an ongoing part of the assessment process, and is necessary to:

- test a candidate's knowledge of facts and procedures
- check if a candidate understands principles and theories *and*
- collect information on the type and purpose of the processes a candidate has gone through
- candidate responses must be recorded

The quality and breadth of evidence provided should determine whether an assessor is confident that a candidate is competent or not. Assessors must be convinced that candidates working on their own can work independently to the required standard.

Additional information

For further information regarding administration for this qualification, please refer to the OCR document '*Admin Guide: Vocational Qualifications*' on the OCR website www.ocr.org.uk .