Unit Title:	**Creating an Object-Oriented Computer Program**	
OCR unit number	13	
Level:	3	
Credit value:	12	
Guided learning hours:	90	
Unit reference number:	L/601/3184	

Candidates undertaking this unit must complete real work activities in a work environment. Simulation is only allowed in exceptional circumstances (please refer to the centre handbook for further details).

Unit purpose and aim

This unit covers more advanced concepts of object-oriented computer languages and their use to implement, refine and test computer programs.

Learning Outcomes	Assessment Criteria	Knowledge, understanding and skills
The Learner will:	**The Learner can:**	Candidates must have an understanding of:
1 Implement a software design using object-oriented programming	1.1 Identify the objects and data and file structures required to implement a given design	• objects
	1.2 Select, declare and initialise variable and data structure types and sizes to implement design requirements	• data
		• file structures
	1.3 Define relationships between objects to implement design requirements	• selecting, declaring and initialising variable and data structure types and sizes to meet given requirements
	1.4 Implement message passing between objects to implement design requirements	• data structure types and sizes
	1.5 Implement object behaviours using control structures to meet the design algorithms	• relationships between objects and how to implement them.
	1.6 Select and declare file structures to meet design file storage requirements	• the implementation of message passing between objects
	1.7 Select and use standard input/output commands to implement design requirements	• object behaviours and the use of control structures to meet design algorithms
	1.8 Make effective use of operators and predefined functions	• how to select and declare file structures to meet file storage requirements
	1.9 Make effective use of an Integrated Development Environment (IDE) including code and screen templates	• standard input/output commands
		• operators and predefined functions
		• IDE including code and screen templates

Learning Outcomes	Assessment Criteria	Knowledge, understanding and skills
2 Refine an object-oriented program to improve quality	2.1 Use an agreed standard for naming, comments and code layout 2.2 Make effective use of encapsulation, polymorphism and inheritance 2.3 Implement data validation for inputs 2.4 Identify and implement opportunities for error handling and reporting	Candidates must have an understanding of: • naming conventions used within programming and be able to explain the key features • the purpose of comments in programming and how to implement them • the appropriate layout for programme code and be able to explain why good layout is important • encapsulation, polymorphism and inheritance and how they are used • a range of data validation methods for inputs • what mechanisms are available for error handling and reporting and be able to implement them
3 Test the operation of an object-oriented driven program	3.1 Make effective use of the debugging facilities available in the IDE 3.2 Prepare a test strategy 3.3 Select suitable test data and determine expected test results 3.4 Record actual test results to enable comparison with expected results 3.5 Analyse actual test results against expected results to identify discrepancies 3.6 Investigate test discrepancies to identify and rectify their causes	Candidates must have an understanding of: • appropriate debugging tools and how they are used within the IDE • how to develop appropriate test plans which will fully test the functionality of the program. They should be able to identify the following: - a range of valid tests - the test data to use - the expected results from the test data • comparing the actual test results with the expected results and: - identify discrepancies - identify how to rectify the problem
4 Document an object-oriented driven program	4.1 Create on-screen help to assist the users of a computer program 4.2 Create documentation for the support and maintenance of a computer program	Candidates must know how to create on-screen help to assist the end-users. Candidates must know how to create documentation for the support and maintenance of the computer program.

Assessment

Candidates undertaking this unit must complete real work activities in order to produce evidence to demonstrate they are occupationally competent. Real work is where the candidate is engaged in activities that contribute to the aims of the organisation by whom they are employed, for example in paid employment or working in a voluntary capacity.

Simulation is only allowed for aspects of units when a candidate is required to complete a work activity that does not occur on a regular basis and therefore opportunities to complete a particular work activity do not easily arise. When simulation is used, assessors must be confident that the simulation replicates the workplace to such an extent that candidates will be able to fully transfer their occupational competence to the workplace and real situations.

Internal quality assurance personnel must agree the use of simulated activities before they take place and must sample all evidence produced through simulated activities.

It is the assessor's role to satisfy themselves that evidence is available for all performance, knowledge and evidence requirements before they can decide that a candidate has finished a unit. Where performance and knowledge requirements allow evidence to be generated by other methods, for example by questioning the candidate, assessors must be satisfied that the candidate will be competent under these conditions or in these types of situations in the workplace in the future. Evidence of questions must include a written account of the question and the candidate's response. Observations and/or witness testimonies must be detailed and put the evidence into context ie the purpose of the work etc.

All of the assessment criteria in the unit must be achieved and clearly evidenced in the submitted work, which is externally assessed by OCR.

Evidence for the knowledge must be explicitly presented and not implied through other forms of evidence.

Evidence requirements

All aspects of the assessment criteria must be covered and evidence must be available that shows where and how the assessment criteria have been achieved.

Assessment Criterion 1

For a given scenario a candidate must identify the functionality of the programme and intended end user.

A plan of their program to include:

• selection, declaration and initialisation of variables

• data and file structure types and sizes including declaration

• objects and the relationships between them

• implementation of message passing between objects

• use of code and screen templates

• input/output commands

• operators and predefined functions

Candidates must create their program design using the code and templates available within their IDE.

Assessment Criterion 2

Candidates must provide:

Printouts of their code confirming that they have used an agreed standard for:

- naming conventions

- using comments

- code layout

An explanation of the following:

- key features of naming conventions

- the purpose of comments in programming

- why good code layout is important

Candidates must identify and implement the following:

- user functions to replace repeating code sequences

- data validation methods for inputs

- error handling and reporting mechanisms

- effective use of encapsulation, polymorphism and inheritance

Assessment Criterion 3

Candidates must:

Identify and use the debugging tools that are available in the IDE.

Prepare detailed test plans identifying and selecting the following:

- a range of valid tests and the associated test data

- expected test results

Implement the test plan and provide evidence of the following:

- actual test results

- analysis of test results against expected results

- plans for the rectification of the problems identified.

Assessment Criterion 4

Candidates must:

Produce program documentation to assist the support and maintenance of the computer program

Provide evidence of the on-screen help they have created which will meet the needs of a variety of end-users.

Candidates are encouraged to choose activities which will allow them to cover all or a majority of the criteria at one time. It is not necessary to use different activities for each element of the criterion.

Guidance on assessment and evidence requirements

Evidence can reflect how the candidate carried out the process or it can be the product of a candidate's work or a product relating to the candidate's competence.

For example: The process that the candidate carries out could be recorded in a detailed personal statement or witness testimony. It is the assessor's responsibility to make sure that the evidence a candidate submits for assessment meets the requirements of the unit.

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

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'

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' (A850)* on the OCR website www.ocr.org.uk .