Unit title:	Object oriented programming
Unit number:	17
Level:	4
Credit value:	15
Guided learning hours:	60
Unit reference number:	K/601/1295

UNIT AIM AND PURPOSE

Learners will be able to design, implement and test an object oriented program. Learners will develop and demonstrate an understanding of good programming principles through designing a system to suit a particular scenario.

LEARNING OUTCOMES AND ASSESSMENT CRITERIA

A pass grade is achieved by meeting **all** the requirements in the assessment criteria.

Learning Outcome (LO) The Learner will:	**Pass** The assessment criteria are the pass requirements for this unit. The Learner can:
LO1 Understand the principles of object oriented programming	1.1 discuss the principles, characteristics and features of objected oriented programming
LO2 Be able to design object oriented programming solutions	2.1 the objects and data and file structures required to implement a given design 2.2 design an object oriented programming solution to a given problem
LO3 Be able to implement object oriented programming solutions	3.1 implement an objected oriented solution based on a prepared design 3.2 define relationships between objects to implement design requirements 3.3 implement object behaviours using control structures to meet the design algorithms 3.4 make effective use of an Integrated Development Environment (IDE), including code and screen templates
LO4 Be able to test and document object oriented programming solutions	4.1 critically review and test an object orientated programming solution 4.2 analyse actual test results against expected results to identify discrepancies 4.3 evaluate independent feedback on a developed object oriented programme solution and make recommendations for improvements

	4.4 create onscreen help to assist the users of a computer program 4.5 create documentation for the support and maintenance of a computer program

GRADING CRITERIA

A merit grade is achieved by meeting **all** the requirements in the pass criteria **and** the merit descriptors.

A distinction grade is achieved by meeting **all** the requirements in the pass criteria **and** the merit descriptors **and** the distinction descriptors.

Merit Criteria (M1, M2, M3) (M1, M2, and M3 are mandatory to achieve a merit grade. Each must be achieved at least once per unit to achieve a merit grade.)	**Distinction Criteria (D1, D2, D3)** (D1, D2, and D3 are mandatory to achieve a distinction grade. Each must be achieved at least once per unit to achieve a distinction grade.) (In order to achieve a distinction grade, all merit criteria must also have been achieved.)
MANDATORY TO ACHIEVE A MERIT GRADE	MANDATORY TO ACHIEVE A DISTINCTION GRADE
M1 Analyse concepts, theories or principles to formulate own responses to situations.	D1 Evaluate approaches to develop strategies in response to actual or anticipated situations.
M2 Analyse own knowledge, understanding and skills to define areas for development.	D2 Evaluate and apply strategies to develop own knowledge, understanding and skills.
M3 Exercise autonomy and judgement when implementing established courses of action.	D3 Determine, direct and communicate new courses of action.

TEACHING CONTENT

The Teaching Content describes what has to be taught to cover **all** Learning Outcomes.

Learners must be able to apply relevant examples to their work although these do not have to be the same as the examples specified.

LO1 Understand the principles of object oriented programming	
Different programming methodologies	Event-driven programming, procedural programming, object oriented programming
Programming principles	Commenting on your code, avoiding code repetition, code efficiency, code consistency, good use of Application Programming Interfaces (APIs), writing maintainable code
Characteristics and features	Classes, objects, fields, constructors, parameters, variables, methods (accessors and mutators), data types, interfaces, inheritance, object interaction.
LO2 Be able to design object oriented programming solutions	
Requirements to implement a design	Data dictionary, variable names, data types, system flowchart, class diagram
Object oriented programming solution	Problem definition, investigation into current systems, definition of end user, requirements specification, acceptance tests, test plans (including test description, expected outcome, actual outcome and improvements made), success criteria, interface design, pseudocode.
LO3 Be able to implement object oriented programming solutions	
Implement a solution	Commenting within the code, efficiency of code, good use of general programming principles, good use of object oriented design principles
Relationships between objects	At least 1 class, multiple objects, fields, constructors, parameters, variables, methods (accessors and mutators)
Control structures	Data types, interfaces, inheritance, object interaction, sensible naming conventions
Use of IDE tools	Features of an IDE include items such as code templates, code tips, help features, project management tools, user interface management, auto complete tools, automatic code generation, warnings and error messages, refactoring code, unit testing, debugging tools, etc.

LO4 Be able to test and document object oriented programming solutions	
Test strategies	Unit testing, system testing, end user testing/acceptance testing, comparing program to success criteria, carrying out test plan
Test planning	Design of a test plan (e.g. purpose of test, expected result, actual result, action required), test data (normal, erroneous, extreme/borderline/boundary)
Analysis of results	Make improvements to a system based on testing results, discussion of the limitations of the system and what improvements could be possible
Onscreen help to assist users	Onscreen help includes items such as error message, helpful dialogues, program specific support, user instructions
Documentation	User guide, training materials, explanation of how the program works, glossary, technical guide, explanation of coding decisions, hardware and software requirements, error handling, troubleshooting.

GUIDANCE

Delivery guidance

It will be beneficial to deliver this unit in a way that uses actual events, industry forecasts or sector specific contexts which offer the learner the opportunity to explore, develop and apply the fundamental principles of the sector or subject area.

Typical delivery contexts could include a booking system, an inventory system, a ticketing system, technical support system, or a game (e.g. a card game, text-based adventure game, accounting software or an invoicing system). There are many different types of object-oriented programming languages and software. As long as the outcome adheres to object-oriented principles and is written in an object-oriented language (such as Python, Java, Delphi, C++, C#, VB.net, etc.), any scenario will meet the requirements of this unit.

Learners will benefit from being encouraged to exercise autonomy and judgement to research programming techniques and then design and implement an object-oriented system.

Assessment evidence guidance
Evidence must be produced to show how a learner has met each of the Learning Outcomes. This evidence could take the form of assignments, project portfolios, presentations or, where appropriate, reflective accounts. As part of their evidence, learners should include screenshots of their test results and their user interface. Learners should include a copy of their source code and a copy of the compiled program, as well as a sources list of their research materials. Where group work/activities contribute to assessment evidence, the individual contribution of each learner must be clearly identified. All evidence must be available for the visiting moderator to review. Where learners are able to use real situations or observations from work placement, care should be taken to ensure that the record of observation accurately reflects the learner's performance. This should be signed, dated, and included in the evidence. It is best practice to record another individual's perspective of how a practical activity was carried out. Centres may wish to use a witness statement as a record of observation. This should be signed and dated and included in the evidence.

RESOURCES

Books
Budd, Timothy., *An Introduction to Object-oriented Programming*, Addison-Wesley; 3rd edition, 2001. Barnes, David., *Objects First with Java: A Practical Introduction Using BlueJ*, Prentice Hall / Pearson Education, 5th edition, 2012. Meyer, Bertrand., *Object-Oriented Software Construction*, Prentice Hall, 2nd edition, 1997.
Websites
There a number of websites will provide relevant information. Learners should be encouraged to research such information rather than have links given to them. Some helpful websites: http://stackoverflow.com/ http://docs.oracle.com/javase/tutorial/ http://docs.python.org/2/tutorial/