

Unit 15: Object Oriented Programming

Unit code:	J/601/7282
QCF Level 3:	BTEC National
Credit value:	10
Guided learning hours:	60

● Aim and purpose

To enable learners to develop the skills and understanding required to design and develop object oriented applications.

● Unit introduction

Object oriented programming is an industry-proven method for developing reliable modular programs and is popular in software engineering. Consistent use of object oriented techniques can lead to shorter development lifecycles, increased productivity and lower the cost of producing and maintaining systems.

Programming with objects simplifies the task of creating and maintaining complex applications. Object oriented programming is a way of modelling software that maps programming code to the real world.

This unit enables learners to become familiar with the underpinning concepts of object oriented programming and subsequently to develop particular skills in an object oriented language. The unit starts by looking at the features of object oriented programming, explores the tools and techniques used in their development and takes learners through design and software development. Learners will use a structured approach to the design and development of applications, ensuring the solution is well documented and tested thoroughly against the original user requirement.

Object orientation is now the cornerstone of many languages; it is dominant in C++, Java, the Microsoft.Net environment and many other systems.

● Learning outcomes

On completion of this unit a learner should:

- 1 Understand the features of object oriented programming
- 2 Be able to use the tools and techniques of an object oriented language
- 3 Be able to design object oriented applications
- 4 Be able to implement object oriented applications.

Unit content

1 Understand the features of object oriented programming

Key features: discrete, reusable units of programming logic; identification of objects; data abstraction; modularity; classification; inheritance; polymorphism; encapsulation; classes; methods; message passing

Programming languages: eg Visual Basic.NET (VB.NET), C++, C#, Java, Python

2 Be able to use the tools and techniques of an object oriented language

Tools: eg predefined functions, screen templates

Techniques: using integrated development environment (IDE)

Variables: global, local, static, overloaded results, instance

3 Be able to design object oriented applications

Classes: class diagram; dependencies and inheritances; identification attributes; methods; the control of scope of attributes and methods; inheritance; aggregation; association; polymorphism; pre-defined classes eg class library, downloaded, imported

Objects: eg constructors, destructors; building a program with reusable objects; defining relationships between objects; implementing message passing between objects

4 Be able to implement object oriented applications

Creation of application: use of development environment; debugging; data validation; error handling and reporting

Programming language syntax: eg selecting, declaring and initialising variable and data structure types and sizes

Constructs: selection eg if ... then ... else, CASE; iteration eg while ... do, repeat ... until

Programming standards: eg use of comments; code layout; indentation

Testing: test strategy; test plan structure eg test, date, expected result, actual result, corrective action; error messages; specialist software tools eg debug

Review: against specifications requirements; interim reviews

Documentation: user; technical

Assessment and grading criteria

In order to pass this unit, the evidence that the learner presents for assessment needs to demonstrate that they can meet all the learning outcomes for the unit. The assessment criteria for a pass grade describe the level of achievement required to pass this unit.

Assessment and grading criteria		
To achieve a pass grade the evidence must show that the learner is able to:	To achieve a merit grade the evidence must show that, in addition to the pass criteria, the learner is able to:	To achieve a distinction grade the evidence must show that, in addition to the pass and merit criteria, the learner is able to:
P1 explain the key features of object oriented programs	M1 explain the importance of encapsulation, polymorphism and inheritance on object oriented programming	D1 evaluate the suitability of object oriented programs for graphical applications
P2 demonstrate the use of object oriented tools and techniques		
P3 design an object oriented application to meet defined requirements [CT1]	M2 give reasons for the tools and techniques used in the production of an object oriented application [IE2, IE6]	
P4 implement a working object oriented application to meet defined requirements [SM2, SM3]		
P5 test an object oriented application [SM4]	M3 analyse actual test results against expected results to identify discrepancies [RL3]	D2 evaluate an object oriented application. [IE4]
P6 create onscreen help to assist the users of a computer program. [RL6]	M4 create technical documentation for the support and maintenance of a computer program. [RL6]	

PLTS: This summary references where applicable, in the square brackets, the elements of the personal, learning and thinking skills applicable in the pass criteria. It identifies opportunities for learners to demonstrate effective application of the referenced elements of the skills.

Key	IE – independent enquirers	RL – reflective learners	SM – self-managers
	CT – creative thinkers	TW – team workers	EP – effective participators

Essential guidance for tutors

Delivery

Learners must have access to facilities that give them the opportunity to evidence all of the criteria fully. If this cannot be guaranteed then centres should not attempt to deliver this unit.

It is likely that, for most learners, this will not be their first contact with formal programming but that it will be their first contact with object oriented programming. It is very important that they use a methodical approach to creating object oriented programs. This will be particularly valuable should learners progress to higher-level courses that involve programming.

A number of small programs should be used to demonstrate object oriented concepts. Any appropriate language can be chosen as the basis of the practical aspects of this unit. An object oriented program should include a simple object oriented design, fully documented class, code, and interfaces (screens, forms, printouts, etc).

It is recommended that learners begin to program early on in the unit. To be of most value, these programming activities should steadily increase in complexity and give lots of opportunity for formative feedback.

Once provided with the task focus, learners should be encouraged to break down the task and submit material in stages for assessment. These stages might naturally be design, documented class, production of program, testing/debugging and documentation.

The suggested delivery pattern follows the order of the learning outcomes. This is not the only sequence that may be used and tutors can follow their own preference.

This unit is not designed with a specific programming language or delivery platform in mind; centres may focus on one or more languages for teaching. The learner will develop an application that must be object oriented and may work on a range of platforms, therefore it may be command line, web based, graphical user-interface based, games-console based or a deliverable for a mobile platform among many other solutions.

Tutors are advised to keep the delivery to one language, although many languages now allow development in multiple platforms.

Learning outcome 1 covers all principles associated with the programming language selected. The advised delivery of the outcome is to cover all programming concepts whilst teaching the concepts of object oriented systems in parallel.

The design in learning outcome 2 may use a range of design methodologies, ensuring that the method selected is suited to the environment selected as well as the programming language of choice.

Implementation in learning outcome 3 must be based on a suitably structured problem that ensures use of more than two modules, and some simple object oriented code, both between events and as a result of events.

Testing in learning outcome 4 must cover the code created for learning outcome 3 and designed in learning outcome 2. The unit software testing can be used to enhance (not replace) this learning outcome and give the learner an extended software development experience.

Whilst this is ideally an introductory unit, developing learning outcomes 2, 3 and 4, learners could be encouraged to devise their own mini-project to develop their higher learning and project management skills in preparation for the working environment.

A centre may select a programming activity of choice, or use an external source (employer, commissioner, open source), the design of the programming solution does not need to be a stand-alone application and may be an enhancement or extension to existing work. Therefore learners completing this unit may contribute to many open source development projects or use them as a basis for their learning experience.

Outline learning plan

The outline learning plan has been included in this unit as guidance and can be used in conjunction with the programme of suggested assignments. The outline learning plan demonstrates one way in planning the delivery and assessment of this unit.

Topic and suggested assignments/activities and/assessment
<p>Introduction to the unit</p> <p>Features of OOP:</p> <ul style="list-style-type: none"> • whole-class exercise – tutor presentation on object oriented programming (what it is, when its used, pros/cons) • whole-class exercise – tutor presentation on the key features of object orientated languages and their role in programming, followed by practical work • whole-class exercise – tutor presentation on an introduction to objects, variables, software engineering, followed by practical work • mixture of tutor-led discussions, practical exercises.
<p>Assignment 1 – How to Start</p> <p>Design methods:</p> <ul style="list-style-type: none"> • whole-class exercise – tutor presentation on program design • directed research – working from tutor-specified sources, classes, inheritance and their role in whole-class exercise – tutor presentation on objects and pre-defined classes, followed by practical work • individual exercise – use tutor-provided exercise materials to better understand simple object oriented code • individual exercise – use tutor-provided exercise materials to better understand the scope of variables • whole-class exercise – tutor presentation on the integration of object oriented code objects • individual exercise – creating an OOP, practising techniques to develop working programs • mixture of tutor-led demonstration, directed learning and learner exercises, delivered over several sessions object oriented driven programming.
<p>Assignment 2 – Design Work</p> <p>Testing/reviewing:</p> <ul style="list-style-type: none"> • whole-class exercise – tutor presentation on debugging, naming conventions etc, followed by individual exercise • whole-class exercise – tutor presentation on testing/reviewing against requirement, followed by practical work • learner exercises.
<p>Assignment 3 – Implement and Test</p>

Topic and suggested assignments/activities and/assessment

Documentation:

- whole-class exercise – tutor presentation on documentation
- individual exercise – following a tutor presentation, prepare different types of support documentation.

Assignment 4 – Finishing Off

Assessment

It is suggested that this unit is assessed using the four assignments summarised in the *Programme of suggested assignments* table.

Finding a scenario which covers all aspects of all criteria is difficult, but the one suggested is acceptable. It places the user in a role which is at an acceptable level for their experience, which is important when devising assignments.

Some of the evidence required to complete the assignments could be naturally occurring within learners' work for other units within the qualification, or for other courses they are undertaking, and tutors are encouraged to use such evidence.

In order to gain a pass grade, learners must achieve all of the pass criteria.

For P1, learners must explain the key features required to implement a given design. This refers to the key features section of the unit content for learning outcome 1. In order to meet this criterion, the learner must describe the features clearly and logically, showing they have recognised the underpinning principles. A presentation or leaflet would be a suitable form of evidence.

For P2, learners must show that they are able to use object oriented programming tools and techniques, including those listed in the unit content for LO2. A presentation would be a suitable form of evidence.

For P3, learners must design an object oriented program. The program needs to be basic only, as suited to the level of learners, but obviously this is at the discretion of the tutor and individual learners. The design should be clear and have no obvious errors.

For P4, learners must create the program they worked on for P3. This program should be fully functional, and fulfil the design aims set down.

For P5, learners must develop and apply an appropriate test plan for the program they worked on for P4. The test plan should test functionality and demonstrate that the program fulfils the design aims and other requirements. Evidence is likely to be a short report on the test plan and results, illustrated with screen grabs.

For P6, learners must produce onscreen help for an object oriented program. The help screens must be coherent and laid out according to the standards that learners have been taught previously. It is up to the tutor at this stage whether they wish to have learners use the work they have done for P4 and P5, or provide learners with a generic object orientated program for which to write the appropriate help.

In order to gain a merit, learners must achieve all of the pass criteria, and all of the merit criteria.

For M1, learners must extend their P1 work to explain the importance of encapsulation, polymorphism and inheritance in object oriented programming. As with P1, evidence for this criterion should be a poster, leaflet or short report would be a suitable evidence, though a presentation can also be used if the learner or tutor would prefer.

For M2, learners must justify their choice of tools and techniques used in the production of the object oriented application created in P3. Evidence for this criterion should be a short report, or similarly detailed presentation.

For M3, learners must analyse the results of their testing in P6. The analysis should compare expected to actual results to identify discrepancies. It would also be expected that learners would suggest what actions should be taken to resolve any problems shown up by the testing. Evidence for this criterion should be a short report. This could be an extension of the P6 report.

For M4, learners will create technical documentation for the support and maintenance of a computer program. As with P6, it is up to the tutor whether they wish to have learners use the work they have carried out for P4 and P5, or provide learners with a generic object orientated program. The documentation must be coherent and laid out according to the standards that learners have previously been taught.

To gain a distinction grade, learners must achieve all of the pass criteria, all the merit criteria and both of the distinction criteria.

Criterion D1 is an extension of P1 and M1, and learners must evaluate the suitability of object oriented programs for graphical applications. Evidence for this criterion should be a short report, or similarly detailed presentation.

For D2, learners will evaluate an object oriented application. There is the option here for the learner to review their P5 work, or to be given a generic program to review. Alternatively, the tutor could give learners the work of one of their peers to review. Evidence for this criterion should be a short report, or similarly detailed presentation.

Programme of suggested assignments

The table below shows a programme of suggested assignments that cover the pass, merit and distinction criteria in the assessment and grading grid. This is for guidance and it is recommended that centres either write their own assignments or adapt any Pearson assignments to meet local needs and resources.

Criteria covered	Assignment title	Scenario	Assessment method
P1, P2, M1, D1	How to Start	You are working as a junior programmer for an electronic games maker. Your managers have asked you to write a short guide to the basics of object oriented programming and to demonstrate some of the techniques.	Leaflet Presentation
P3, M2,	Design Work	You have been asked to design an example object oriented program to show off your coding skills.	Practical work Report
P4, P5, M3, D2	Implement and Test	Your design has been approved and you are now asked to implement and test the program.	Practical work
P6, M4	Finishing Off	Your managers would now like you to show how to write a set of support documentation for a program.	Practical work Short report

Links to other BTEC units

This unit forms part of the BTEC in IT sector suite. This unit has particular links with the following unit titles in the IT suite:

Level 1	Level 2	Level 3
		Unit 6: Software Design and Development
		Unit 15: Object Oriented Programming
		Unit 16: Procedural Programming
		Unit 22: Developing Computer Games

Essential resources

Learners will need individual access to an appropriate development environment according to the particular language chosen by the centre.

Learners will require access to computer equipment to enable them to gain a practical awareness and enable them to apply their knowledge and understanding in a practical situation.

Employer engagement and vocational contexts

The use of vocational context is essential in the delivery and assessment of this unit.

There are a range of organisations that may be able to help to centres engage and involve local employers in the delivery of this unit, for example:

- Learning and Skills Network
- Network for Science, Technology, Engineering and Maths Network Ambassadors Scheme
- National Education and Business Partnership Network
- Local, regional Business links
- Work-based learning guidance.

Delivery of personal, learning and thinking skills

The table below identifies the opportunities for personal, learning and thinking skills (PLTS) that have been included within the pass assessment criteria of this unit.

Skill	When learners are ...
Creative thinkers	<p>generating ideas and exploring possibilities to create a simple object oriented design for a program</p> <p>generating ideas and exploring possibilities to create classes to be used in a program</p> <p>generating ideas and exploring possibilities to create new objects from pre-defined classes in a program</p>
Reflective learners	<p>communicating their learning by creating onscreen help to assist users of a computer program</p>
Self-managers	<p>working towards goals, showing initiative, commitment and perseverance when developing an object oriented application.</p> <p>organising time and resources when developing an object oriented application</p> <p>anticipating, taking and managing risks when testing an object oriented application.</p>

Although PLTS are identified within this unit as an inherent part of the assessment criteria, there are further opportunities to develop a range of PLTS through various approaches to teaching and learning.

Skill	When learners are ...
Independent enquirers	<p>planning and carrying out research on an object oriented application, appreciating the consequences of decisions</p> <p>supporting conclusions when giving reasons for the tools and techniques used in the production of an object oriented application, using reasoned arguments and evidence</p> <p>analysing and evaluating an object oriented application, judging its relevance and value</p>
Reflective learners	<p>reviewing progress on an object oriented application, acting on the outcomes</p> <p>communicating their learning by creating technical documentation for the support and maintenance of a computer program.</p>

● Functional Skills – Level 2

Skill	When learners are ...
ICT – Using ICT	
Plan solutions to complex tasks by analysing the necessary stages	designing an object oriented application
Select, interact with and use ICT systems safely and securely for a complex task in non-routine and unfamiliar contexts	implementing an object oriented application
ICT – Finding and selecting information	
Use appropriate search techniques to locate and select relevant information	explaining the key features of object oriented programs
ICT – Developing, presenting and communicating information	
Enter, develop and refine information using appropriate software to meet requirements of a complex task	implementing an object oriented application
Combine and present information in ways that are fit for purpose and audience	creating onscreen help and technical documentation to assist users, and to provide support of a computer program
Evaluate the selection, use and effectiveness of ICT tools and facilities used to present information	giving reasons for the tools and techniques used in the production of an object oriented application
Mathematics – Representing	
Understand routine and non-routine problems in familiar and unfamiliar contexts and situations	designing and implementing an object oriented application
Identify the situation or problems and identify the mathematical methods needed to solve them	designing and implementing an object oriented application
Choose from a range of mathematics to find solutions	designing and implementing an object oriented application.