Unit 19: Object Oriented Programming

Unit code: K/601/5105

QCF Level 2: BTEC First

Credit value: 10

Guided learning hours: 60

Aim and purpose

To introduce learners to the features of object oriented programming and the object oriented development environment and to develop the skills required to design, implement, test, document and review object oriented programs.

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 life-cycles, increased productivity and therefore 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. Object orientation is now the cornerstone of many languages; it is dominant in Visual Basic, C++, Java, the Microsoft .Net environment, Action Script and many other systems.

This unit starts by looking at the features of object oriented programming and the object oriented environment. Learners will discuss how these types of language are used and their suitability for different uses. The unit then puts theory into practise and learners will design and develop object oriented programs to meet specified needs. There are many concepts to be learned and practised to enable learners to confidently design and implement programs.

As with all programming, a focus on developing solutions to meet identified needs is made, along with an emphasis on the importance of testing and reviewing. Good practice in testing and documenting programs will be emphasised and learners will develop techniques to ensure their program code is understandable and able to be maintained by others.

Learning outcomes

On completion of this unit a learner should:

1 Know the features of object oriented programming

2 Understand the object oriented development environment

3 Be able to design and implement object oriented programs

4 Be able to test, document and review object oriented programs.

Unit content

1 Know the features of object oriented programming

Features: key features eg discrete and reusable units of programming logic, identification of objects, data abstraction, modularity, classification, inheritance, polymorphism, encapsulation, classes, methods, message passing, flexibility, suitability for graphical user interfaces (GUI), simplicity of programming, ease of development

Programming languages: eg Visual Basic.NET (VB.NET), C#, Java, Coldfusion, Perl, PHP (Hypertext Preprocessor)

2 Understand the object oriented development environment

Environment components: IDE; tool boxes and controls; toolbars; predefined functions; screen templates; help menus; debugging tools

3 Be able to design and implement object oriented programs

Design: classes; objects; methods; control structures; data structures; data validation; error handling and reporting

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

Control structures: selection eg if …then…else, CASE; iteration eg while…do, repeat…until

Data structures: selecting, declaring and initialising variable and data structure types and sizes; operators eg logical, arithmetic, relational

Data validation: eg type, range, presence, format

Implement: program and debug

Error handling: debugging eg*:* compiler/translator; errors eg logical, syntax, run-time

4 Be able to test, document and review object oriented programs

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

Documentation: listing of code; support and maintenance documentation

Programming standards: use of comments; code layout; indentation

Review: against specifications requirements; suggest improvements

Assessment and grading criteria

In order to pass this unit, the evidence that learners present 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	identify the key features of object oriented programming	**M1**	describe the features of an object oriented language that make it suitable for creating a GUI	**D1**	evaluate the suitability of object oriented programs for non-graphical applications [IE4]
P2	explain how development environment components simplify the development process				
P3	design and implement an object oriented program to meet defined requirements [IE1, CT1, CT6, SM2, SM3]	**M2**	implement data validation for inputs [CT1]	**D2**	implement error handling and reporting.
P4	test an object oriented program [SM4]	**M3**	compare actual test results against expected results to identify discrepancies.		
P5	create documentation for the support and maintenance of an object oriented program				
P6	review an object oriented program, suggesting possible improvements. [EP4]				

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

It is likely that, for most learners, this will not be their first contact with formal programming but 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 delivery of this unit. To be most valuable, these programming activities should steadily increase in complexity and provide lots of opportunity for formative feedback.

Once they have been given the focus of an assessed task, the 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.

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
Introduction to the unit
The features of OOP: • whole-class exercise – tutor presentation on object oriented programming (what it is, when it is used, pros/cons) • whole-class exercise – tutor presentation on objects, variables, software engineering. Followed by individual exercise.
Design methods: • whole-class exercise – tutor presentation on classes, inheritances etc. Followed by individual exercise • whole-class exercise – tutor presentation on objects, pre-defined classes. Followed by individual exercise • mixture of tutor-led demonstration and learner exercises, repeated over a number of sessions.
Assignment 1 – OOPs
Creating an OO program: • whole-class exercise – tutor presentation on techniques to develop working programs, followed by individual exercise • mixture of tutor-led demonstration and learner exercises, repeated over a number of sessions.

Topic and suggested assignments/activities and assessment
Assignment 2 – Designing an OOP
Testing/reviewing: • whole-class exercise – tutor presentation on debugging, naming conventions etc, followed by individual exercise • whole-class exercise – tutor presentation on reviewing against requirement, followed by individual exercise • mixture of tutor-led demonstration and learner exercises, repeated over a number of sessions.
Assignment 3 – Implementing an OOP

Assessment

It is suggested that this unit is assessed using three assignments as summarised in the *Programme of suggested assignments* table which follows this guidance.

The programme of suggested assignments table suggests the theoretical elements of the assessment (P1, P2, M1 and D1) are covered by producing a booklet, but these could be evidenced using a presentation or any other suitable format.

To achieve a pass grade, learners must achieve the pass criteria listed in the assessment and grading criteria grid.

P1 requires learners to identify the key features of object oriented languages. A written report is probably the easiest format to generate evidence but a presentation or any other suitable method can be used.

For P2, learners must explain how environment components make the development process easier. This requires more than a straightforward explanation of each element outlined in the unit content, the emphasis being on their purpose.

For P3, learners should be guided to create some form of basic object oriented design. A simplified class diagram may be used showing the class attributes, methods and its association with other classes. Class-responsibility-collaboration (CRC) cards are also suitable. Centres may also incorporate use of case diagrams and sequence diagrams if they feel this is appropriate. If learners use a graphic user interface (GUI) with their program then it is important that they show evidence of the planning and design of the GUI interface at an early stage.

For P3 and P4, before and after screenshots of the implemented program design will demonstrate debugging. Witness statements may be used to assess the completed design. For P4, test plans and results are also required and evidence of dealing with error messages.

For P5, documentation should include annotated and properly laid out code (to programming standards) along with the technical documentation necessary to support and maintain the program.

P6 covers the review process and should include suggestions for improvements. This can be included with the documentation for P5.

To achieve a merit grade, learners must achieve the pass grade criteria and the merit grade criteria.

For M1, learners must describe the features that make an object oriented language suitable for creating a GUI. This does not relate to a specific language.

For M2, learners must show evidence of implementing data validation on input.

For M3, a test plan showing expected test results will need to be provided along with the actual test results.

To achieve a distinction grade learners must achieve all the pass and merit criteria and the distinction grade criteria.

For D1, learners must evaluate the suitability of using object oriented programs for non-graphical applications. As with all evaluations this should consider the pros and the cons.

For D2, learners should provide evidence of error handling and reporting.

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 Edexcel assignments to meet local needs and resources.

Criteria covered	Assignment title	Scenario	Assessment method
P1, P2, M1, D1	OOPs!	You are to produce a booklet for trainee programmers on the basics of object oriented programming.	Booklet
P3, M2,D2	Designing an OOP	You are to design and implement a program to a given specification.	Design documentation Screenshots Witness statement
P4, P5, P6, M3	Implementing an OOP	You are to test and document the program.	Documentation Screenshots, program code etc Test plan Witness statement Review

Links to National Occupational Standards, other BTEC units, other BTEC qualifications and other relevant units and qualifications

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 15: Object Oriented Programming

This unit maps to some of the underpinning knowledge from the following areas of competence in the Level 2 National Occupational Standards for IT (ProCom):

- 6.1 Information Management.

Essential resources

Learners will need individual access to an appropriate development environment according to the particular language chosen by the centre. In addition, it would be valuable for perhaps a limited number of additional copies to be made available in open areas, such as learning resources or libraries, so that learners can practise and create assignment material outside of normal contact time.

Employer engagement and vocational contexts

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

There is 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 – www.vocationallearning.org.uk
- Local, regional business links – www.businesslink.gov.uk
- National Education and Business Partnership Network – www.nebpn.org
- Network for Science, Technology, Engineering and Maths Network Ambassadors Scheme – www.stemnet.org.uk
- Work-based learning guidance – www.aimhighersw.ac.uk/wbl.htm
- Work Experience/Workplace learning frameworks – Centre for Education and Industry (CEI University of Warwick) – www.warwick.ac.uk/wie/cei.

Indicative reading for learners

Textbooks

Halvorson V – *Visual Basic 2005 Step by Step* (Microsoft Press US, 2005) ISBN 0735621314

Henney K and Templeman J – *Microsoft Visual C++ .NET Step by Step: Version 2003* (Microsoft Press US, 2003) ISBN 0735619077

Kaldahl B – *EZ Flash MX: Animation, Action Script and Gaming for Macromedia Flash* (Trafford Publishing, 2004) ISBN 1412006171

Lemay L and Cadenhead R – *Sams Teach Yourself Java 2 in 21 Days, 4th Edition* (Sams, 2004) ISBN 0672326280

Schildt H – *C++: A Beginner's Guide, 2nd Edition* (McGraw-Hill Education, 2003) ISBN 0072232153

Websites

http://java.sun.com/docs/books/tutorial/java/concepts/

www.oopweb.com/

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 …
Independent enquirers	identifying questions to answer and problems to resolve when designing and implementing an object oriented program to meet defined requirements
Creative thinkers	generating ideas and exploring possibilities to create a simple object oriented design for a program, and adapting ideas as circumstances change generating ideas and exploring possibilities to create classes to be used in a program, and adapting ideas as circumstances change generating ideas and exploring possibilities to create new objects from pre-defined classes in a program, and adapting ideas as circumstances change
Self-managers	working towards goals, showing initiative, commitment and perseverance when design and implementing an object oriented program organising time and resources when designing and implementing an object oriented program anticipating, taking and managing risks when testing an object oriented program
Effective participators	identifying improvements when reviewing an object oriented program.

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	analysing and evaluating the suitability of object oriented programs for non-graphical applications
Creative thinkers	generating ideas and exploring possibilities when implementing data validation for inputs.

Functional Skills – Level 2

Skill	When learners are …
ICT – Using ICT	
Select, interact with and use ICT systems safely and securely for a complex task in non-routine and unfamiliar contexts	implementing a program
ICT – Developing, presenting and communicating information	
Combine and present information in ways that are fit for purpose and audience	creating documentation for maintenance and support
Evaluate the selection, use and effectiveness of ICT tools and facilities used to present information	reviewing the program and suggesting improvements.