

## Unit 41: Programming in Java

**Unit code:** F/601/1528

**QCF Level 5:** BTEC Higher National

**Credit value:** 15

### ● Unit aim

To provide learners with an understanding of the principles of programming in Java, exploring the object oriented nature of the language and the multi-platform versatility offered.

### ● Unit abstract

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

Java is synonymous with the object orient paradigm offering all the features of the technology in a format that can be used on many differing systems. The development of systems with Java objects simplifies the task of creating and maintaining complex applications.

Many environments use Java as its 'underpinning' framework, with Java applications found on mobile phones, dedicated systems, web-based multimedia, security and control systems as well as traditional applications and bespoke operating systems.

Learners taking this unit will have the opportunity to develop their understanding of the Java programming language and develop code suited to a range of platforms. The unit is not specific to one instance of the Java programming language and may be used to deploy, among others, mobile applications, bespoke applications or web-based solutions.

### ● Learning outcomes

**On successful completion of this unit a learner will:**

- 1 Understand the principles of programming in Java
- 2 Be able to design Java solutions
- 3 Be able to implement Java solutions
- 4 Be able to test and document Java solutions.

## Unit content

---

### 1 Understand the principles of programming in Java

*Characteristics:* Java Virtual Machine (JVM); Java platform; classed-based; object-oriented; compilers; class libraries; applications; applets; object models; enforced error handling; concurrency; threads, multi-platform

*Reasons for choice of language:* organisational policy; suitability of features and tools; availability of trained staff; reliability; development and maintenance costs; expandability

*Object models:* inheritance; polymorphism; encapsulation; public classes; private classes; public methods; private methods

*Data structures:* public instance variables; private instance variables; naming conventions; arrays (one-dimensional, two-dimensional); file structures; loops eg conditional (pre-check, post-check, break-points), fixed; conditional statements; case statements; logical operators; assignment statements; input statements; output statements

*Data types:* constants and literals; integer; floating point; byte; date; boolean; others eg character, string, small int; choice of data types eg additional validation, efficiency of storage

*Environment:* features eg interpreted, run time environment, system specific libraries

*Programming syntax:* features eg command rules, variable declaration, class/method declaration

*Standards:* features eg use of comments, code layout, indentation

### 2 Be able to design Java solutions

*Requirements specification:* overview eg inputs, outputs, processing, user interface; constraints eg hardware platforms, timescales for development; delivery environment eg mobile, hand-held, web based, desktop; interaction eg data exchange, compliance, compatibility, standards

*Program design:* tools eg structure diagrams, data flow diagrams, entity relationship models, flow charts, pseudo code, class diagrams, class responsibilities, collaboration cards; inheritance

*Technical documentation:* requirements specification; others as appropriate to language eg form design, flowcharts, pseudo code, structured English, action charts, data dictionary, class and instance diagrams

### 3 Be able to implement Java solutions

*Classes:* features eg identification attributes, methods, control of scope of attributes and methods, inheritance, aggregation, association, polymorphism

*Programming:* use of conventional language commands; use of library classes; pre-defined eg class libraries, downloaded, imported

*Complexity:* multiple classes; inheritance; reuse of objects; application of polymorphism

### 4 Be able to test and document Java solutions

*Mechanisms:* methods eg valid declarations, debugging code, checking naming conventions, checking functionality against requirements, error detection, error messages, compiler errors, runtime errors, in code response, dry running

*Feedback:* record feedback, eg surveys, questionnaire, interviews; analyze feedback; present results

*Supportive documentation:* test plan; test results; programmer guidance; user guidance

*Review:* design against specification requirements, interim reviews

## Learning outcomes and assessment criteria

Learning outcomes  On successful completion of this unit a learner will	Assessment criteria for pass  The learner can
LO1 Understand the principles of programming in Java	1.1 discuss the principles, characteristics and features of programming in Java 1.2 critically evaluate the environmental flexibility of programming in Java
LO2 Be able to design Java solutions	2.1 design a Java programming solution to a given problem 2.2 explain the components and data and file structures required to implement a given design
LO3 Be able to implement Java solutions	3.1 implement a Java programming 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 identify and implement opportunities for error handling and reporting 3.5 make effective use of an Integrated Development Environment (IDE) including code and screen templates
LO4 Be able to test and document Java solutions	4.1 critically review and test a Java programming solution 4.2 analyse actual test results against expected results to identify discrepancies 4.3 evaluate independent feedback on a developed Java program solution and make recommendations for improvements 4.4 create user documentation for the developed Java program solution 4.5 create technical documentation for the support and maintenance of a Java program solution.

## Guidance

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

The learning outcomes associated with this unit are closely linked with:

Level 3	Level 4	Level 5
Unit 6: Software Design and Development	Unit 18: Procedural Programming	Unit 39: Computer Games Design and Development
Unit 14: Event Driven Programming	Unit 19: Object Oriented Programming	Unit 40: Distributed Software Applications
Unit 15: Object Oriented Programming	Unit 20: Event Driven Programming Solutions	Unit 42: Programming in .NET
Unit 16: Procedural Programming	Unit 21: Software Applications Testing	
	Unit 22: Office Solutions Development	
	Unit 23: Mathematics for Software Development	

This unit has links to the Level 4 and Level 5 National Occupational Standards for IT and Telecoms Professionals, particularly the areas of competence of:

- Software Development.

### Essential requirements

Whilst some procedural languages are commercially available, there are also free languages available incorporating a diverse range of commands, commonly deployed on many platforms. Centres must ensure that in the case of mobile platforms, the applicable free emulators are available.

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

Learners must develop an application that may be event driven, an applet, or command line driven and it may work on a range of platforms. It may be web based, GUI based, a games console or a deliverable for a mobile platform amongst many other solutions.

Centres must 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 must be based on a suitably structured problem that ensures the use of Java elements. Centres must select a programming activity, or use an external source (employer, commissioner, open source).

## Resources

### Books

Bloch J – *Effective Java, Second Edition* (Prentice Hall, 2008) ISBN: 0321356683

Goetz B – *Java Concurrency in Practice* (Addison Wesley, 2006) ISBN: 0321349601

Niemeyer P – *Learning Java, Third Edition* (O'Reilly, 2005) ISBN: 0596008732

### Websites

<http://java.sun.com/docs/books/tutorial/>

<http://math.hws.edu/javanotes/>

[www.idevelopment.info/data/Programming/java/PROGRAMMING\\_Java\\_Programming.shtml](http://www.idevelopment.info/data/Programming/java/PROGRAMMING_Java_Programming.shtml)

## Employer engagement and vocational contexts

Working with a local programming-based organisation or using internet-based open source projects would enhance the learners' experience and offer a relevant vocational context.

## Unit 42: Programming in .NET

**Unit code:** H/601/1537

**QCF Level 5:** BTEC Higher National

**Credit value:** 15

### ● Aim

To provide learners with an understanding of the principles of programming using a .NET framework as an underpinning technological concept in the fields of programming and systems development.

### ● Unit abstract

The .NET framework defines a range of reusable class libraries that define the interactions used for Windows operating systems based development of utilities, applications, web based resources, games as well as data integration.

Whilst specific to Microsoft products, the .NET framework defines interactions with servers, workstations and mobile devices. The .NET framework also describes interactions and data exchange with other programming and development systems and is designed to enable cross-platform interaction.

This unit allows learners to become familiar with the underpinning concepts of .NET framework programming, without needing to develop particular skills in one chosen language. Each of the languages has the capacity to develop event driven solutions and it is not important which language is chosen as long as the skills being developed and evidenced relate to the key .NET focus.

The focus of the unit is on developing solutions to meet identified user needs while emphasising the importance of testing and reviewing.

### ● Learning outcomes

**On successful completion of this unit a learner will:**

- 1 Understand the principles of programming using a .NET framework.
- 2 Be able to design .NET solutions
- 3 Be able to implement .NET solutions
- 4 Be able to test and document .NET solutions.