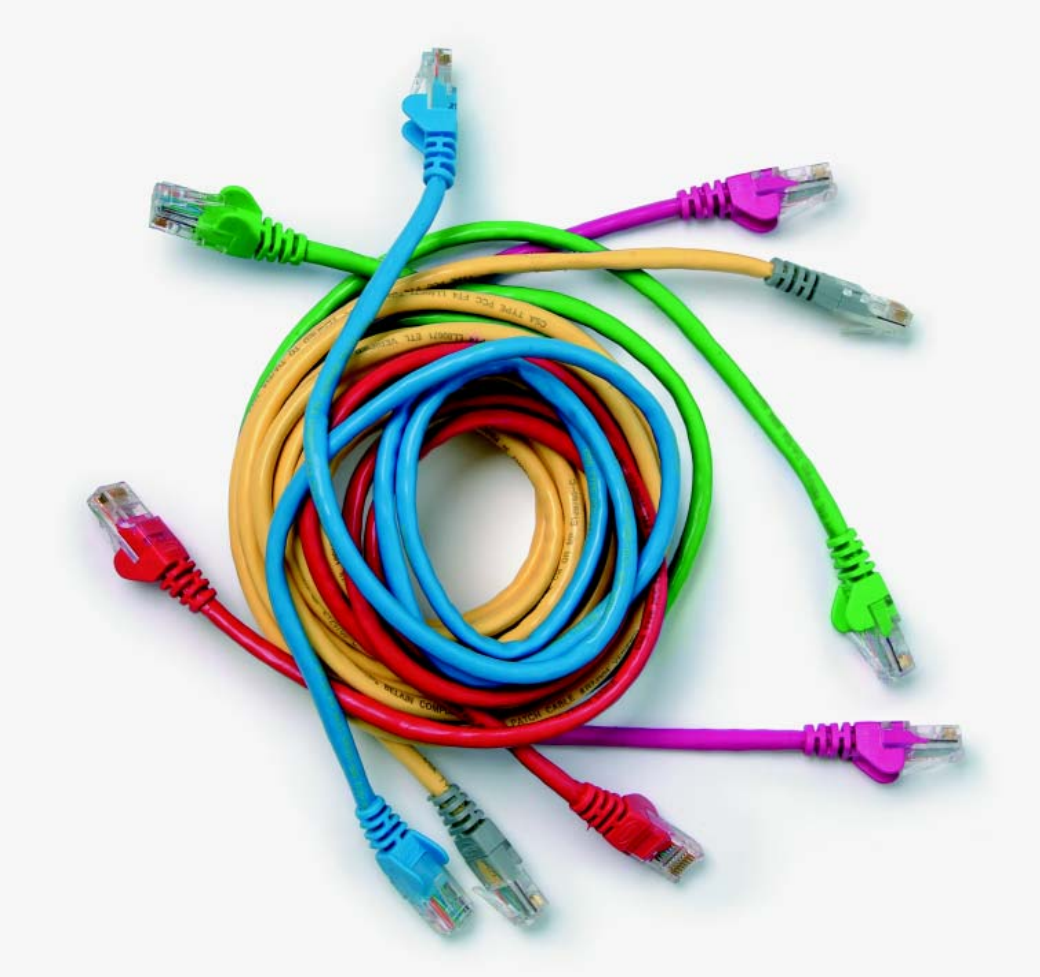
# Systems and Principles Unit Syllabus

## Level 3 Creating an object oriented computer program using Java

7540-037

www.cityandguilds.com
October 2010
Version 2.0



**About City & Guilds**
City & Guilds is the UK's leading provider of vocational qualifications, offering over 500 awards across a wide range of industries, and progressing from entry level to the highest levels of professional achievement. With over 8500 centres in 100 countries, City & Guilds is recognised by employers worldwide for providing qualifications that offer proof of the skills they need to get the job done.

**City & Guilds Group**
The City & Guilds Group includes City & Guilds, City & Guilds Institute, ILM (the Institute of Leadership & Management) which provides management qualifications, learning materials and membership services, NPTC which offers land-based qualifications and membership services, and HAB (the Hospitality Awarding Body). City & Guilds also manages the Engineering Council Examinations on behalf of the Engineering Council.

**Equal opportunities**
City & Guilds fully supports the principle of equal opportunities and we are committed to satisfying this principle in all our activities and published material. A copy of our equal opportunities policy statement is available on the City & Guilds website.

**Copyright**
The content of this document is, unless otherwise indicated, © City & Guilds Limited and may not be copied, reproduced or distributed without prior written consent.

However, approved City & Guilds centres and candidates studying for City & Guilds qualifications may photocopy this document free of charge and/or include a locked PDF version of it on centre intranets on the following conditions:

- centre staff may copy the material only for the purpose of teaching candidates working towards a City & Guilds qualification, or for internal administration purposes
- candidates may copy the material only for their own use when working towards a City & Guilds qualification

The *Standard Copying Conditions* on the City & Guilds website also apply.

Please note: National Occupational Standards are not © City & Guilds Limited. Please check the conditions upon which they may be copied with the relevant Sector Skills Council.

**Publications**
City & Guilds publications are available on the City & Guilds website or from our Publications Sales department at the address below or by telephoning +44 (0)20 7294 2850 or faxing +44 (0)20 7294 3387.

Every effort has been made to ensure that the information contained in this publication is true and correct at the time of going to press. However, City & Guilds' products and services are subject to continuous development and improvement and the right is reserved to change products and services from time to time. City & Guilds cannot accept liability for loss or damage arising from the use of information in this publication.

**City & Guilds**
**1 Giltspur Street**
**London EC1A 9DD**
**T +44 (0)844 543 0000 (Centres)**
**T +44 (0)844 543 0033 (Learners)**
**F +44 (0)20 7294 2413**
**www.cityandguilds.com**
**learnersupport@cityandguilds.com**

# Contents

## Unit 037 Creating an object oriented computer program using Java


| | | |
|---|---|---|
| Syllabus Overview | | 2 |
| Outcome 1 | Implement a software design using object oriented programming | 3 |
| Outcome 2 | Refine an object oriented program to improve quality | 4 |
| Outcome 3 | Test the operation of an object oriented program | 5 |
| Outcome 4 | Document an object oriented program | 6 |
| Unit record sheet | | 8 |

# Unit 037 Creating an object oriented computer program using Java

Syllabus Overview

**Unit accreditation number L/601/3184**

**Credit value 12**

**Rationale**

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

**Learning outcomes**

There are **four** outcomes to this unit. The candidate will be able to:

1 Implement a software design using object oriented programming
2 Refine an object oriented program to improve quality
3 Test the operation of an object oriented program
4 Document an object oriented program

**Guided learning hours**

It is recommended that **90** guided learning hours should be allocated for this unit. This may be on a full time or part time basis.

**Connections with other qualifications**

This unit contributes towards the learning outcomes and assessment criteria required for the level 3 Diploma in ICT Professional Competence.

**Assessment and grading**

Assessment will be by means of a **set assignment** covering practical activities and underpinning knowledge.

# Unit 037 Creating an object oriented computer program using Java

## Outcome 1 Implement a software design using object oriented programming

**Practical activities**

The learner will be able to

1 identify the objects and data and file structures required to implement a given design
2 select, declare and initialise variable and data structure types and sizes to implement design requirements
3 define relationships between objects to implement design requirements
4 implement message passing between objects to implement design requirements
5 implement object behaviours using control structures to meet the design algorithms
6 select and declare file structures to meet design file storage requirements
7 select and use standard input/output commands to implement design requirements
8 make effective use of operators and predefined functions
9 make effective use of an Integrated Development Environment (IDE) including code and screen templates

**Underpinning knowledge**

The learner will be able to

- describe the data types: byte, int, float, double, boolean and char and the data qualifiers long, short, signed and unsigned
- describe the use of String and StringBuffer classes to manipulate strings
- explain the use of arrays of data type: byte, int, char, float. double
- describe the use of header files to define constants and external functions
- explain that data within an object cannot be accessed directly but is accessed through its interface
- explain that an object's interface is the functions and parameters defined within the object that can be accessed by other objects
- describe how message passing between objects means that a function within one object is called by another object and data is passed through the function's defined parameters
- describe how a value can be returned by a called function
- explain the use of the private and public qualifiers
- explain the purpose of overloading methods
- describe control structures used for loops ie while, do … while, for
- describe control structures used for selection ie if, if … else, switch
- explain the use of packages and the purpose of the import statement
- describe the use of System.in.read() for standard input and System.out.print() and System.out.println() for standard output
- describe the relational operators: < (less than), > (greater than), <= (less than or equal to), >= (greater than or equal to), == (equal to), != (not equal to)
- describe the logical operators: ! (not), && (and), || (or)
- explain the use of the increment (++) and decrement (--) operators in prefix and postfix mode
- describe the arithmetic operators ie * (multiply), / (divide), - (subtract), + (add), % (modulus)
- state the difference between the assignment operator (=) and the relational operator (==)
- describe the use of the assignment operators: +=, -=, *=, /=, %=

# Unit 037 Creating an object oriented computer program using Java

## Outcome 2 Refine an object oriented program to improve quality

**Practical activities**

The learner will be able to

1 use an agreed standard for naming, comments and code layout
2 make effective use of encapsulation, polymorphism and inheritance
3 implement data validation for inputs
4 identify and implement opportunities for error handling and reporting

**Underpinning knowledge**

The learner will be able to

- describe the conventional use of indentation in code layout
- state that meaningful names should be used for objects, variables and functions and that constants are normally identified using uppercase characters and variables using lowercase characters
- state that meaningful comments are inserted in code to aid understanding of the code
- describe Java inheritance for a class and the use of **this** and **extends**
- state that data validation is performed on data entered into a program to prevent incorrect data causing incorrect results or a run-time error
- describe the types of data validation that can be performed such as presence check, range check, date check, type check (alphabetic or numeric), character count, check digit (modulus number), format check (eg AG145), use of a lookup table for defined values
- state the importance of trapping errors in a program so that the program does not crash at run-time
- describe the use of try … catch … finally to trap errors
- describe how screen prompts are used to provide information to a user about the actions that can be taken after an error

# Unit 037 Creating an object oriented computer program using Java

## Outcome 3 Test the operation of an object oriented program

**Practical activities**

The learner will be able to

1 make effective use of the debugging facilities available in the IDE
2 prepare a test strategy
3 select suitable test data and determine expected test results
4 record actual test results to enable comparison with expected results
5 analyse actual test results against expected results to identify discrepancies
6 investigate test discrepancies to identify and rectify their causes

**Underpinning knowledge**

The learner will be able to

- state that errors can be located when debugging a program by displaying the values held in variables
- explain the purpose of a test plan is, for each test to be performed, to identify the type of test, the test data required and the expected results of the test
- state that test data should contain valid and invalid data
- explain the purpose of a test log is to record the actual results of each test in the test plan, comment on any discrepancies between the actual results and the expected results and record if any amendments are made to correct an error
- state that testing is done to determine if a program executes correctly according to its specification and to aid in the location and correction of errors

# Unit 037 Creating an object oriented computer program using Java

## Outcome 4 Document an object oriented program

**Practical activities**

The candidate will be able to

1 create on-screen help to assist the users of a computer program

2 create documentation for the support and maintenance of a computer program

**Underpinning knowledge**

The learner will be able to

- describe how screen prompts are used to provide information to a user about the actions that can be taken
- state that the purpose of technical documentation is to help the software developer support and maintain the software
- describe the contents of technical documentation ie program specification, program listing, class interfaces, test plan and test results

# Unit record sheet

Use this form to track your progress through this unit.

Tick the boxes when you have covered each outcome. When they are all ticked, you are ready to be assessed.

| Outcome | ✓ | Date |
|---|---|---|
| 1 Implement a software design using object oriented programming | ☐ | |
| 2 Refine an object oriented program to improve quality | ☐ | |
| 3 Test the operation of an object oriented program | ☐ | |
| 4 Document an object oriented program | ☐ | |

**Candidate Signature** ..... **Date** .....

**City & Guilds Registration Number** .....

**Quality nominee (if sampled)** ..... **Date** .....

**Assessor Signature** ..... **Date** .....

**External Verifier Signature (if sampled)** ..... **Date** .....

**Centre Name** ..... **Centre Number** .....

**Published by City & Guilds**
**1 Giltspur Street**
**London**
**EC1A 9DD**
**T +44 (0)844 543 0000 (Centres)**
**T +44 (0)844 543 0033 (Learners)**
**F +44 (0)20 7294 2400**
**www.cityandguilds.com**

City & Guilds is a company registered in England and Wales (company number 16513878)