e-Quals Unit Syllabus

Level 3 Develop designs and test software components

7266/7267-301

www.cityandguilds.com/e-quals07
August 2008
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, 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 *Access to assessment and qualifications* is available on the City & Guilds website.

Copyright

The content of this document is, unless otherwise indicated, © The City and Guilds of London Institute 2007 and may not be copied, reproduced or distributed without prior written consent.

However, approved City & Guilds centres and learners 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 learners working towards a City & Guilds qualification, or for internal administration purposes
- learners 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.

Please note: National Occupational Standards are not © The City and Guilds of London Institute. 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)20 7294 2787
F +44 (0)20 7294 2413
www.cityandguilds.com
centresupport@cityandguilds.com

Contents

Unit 301	**Develop designs and test software components**	
Outcome 1	Explain the concepts of software design	4
Outcome 2	Use diagrammatical representation for design	5
Outcome 3	Specify data types and data structures	6
Outcome 4	Develop a software design specification	8
Outcome 5	Plan for software testing	10
Outcome 6	Record and analyse test results	12
Outcome 7	Identify Health and Safety requirements	14
Unit record sheet		15
Appendix A		16
Appendix B		17
Appendix C		18

Unit 301 Develop designs and test software components

Rationale

The aim of this unit is to enable candidates to develop the skills required to create a detailed software design specification from a given outline program design specification and to plan for and test software systems.

The use of computerised tools for producing Gantt and PERT charts (project management software) is preferred **but is not a mandatory requirement for the completion of this unit.**

Learning outcomes

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

Explain the concepts of software design

Use diagrammatical representation for design

Specify data types and data structures

Develop a software design specification

Plan for software testing

Record and analyse test results

Identify Health and Safety requirements

Guided learning hours

It is recommended that 60 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 knowledge and understanding required for the following qualifications:

Outcome	This award contributes to the knowledge and understanding of the following Areas of Occupational Competence in the City & Guilds NVQ for IT Professionals (4324)
1,2,3,4	312 Software development – design 3
5,6	319 Testing ICT systems 3
7	301 Health and Safety in ICT and Contact Centres 3

Key Skills

This unit contributes towards the Key Skills in the following areas:

Communication	C3.2, C3.3
Application of Number	N1.1
Information technology	None
Working with others	None
Improving own learning	LP3.1, LP3.2, LP3.3
Problem solving	PS3.1, PS3.2, PS3.3

Assessment and grading

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

Unit 301 Develop designs and test software components

Outcome 1 Explain the concepts of software design

Underpinning knowledge

The candidate will be able to:

1 describe the advantages in adopting a formal design procedure:

- a maintainability of code
- b reliability of code
- c reusability of code
- d comprehension of code
- e hardware independence (portability of code)
- f code is easily checked and tested
- g language independence

2 list and describe the three fundamental program constructs:

- a sequence
- b selection
- c iteration

3 describe and distinguish between various types of iteration:

- a definite
- b indefinite
- c pre-condition
- d post-condition

4 describe the purpose and value of using a structured approach by transferring control to

- a functions
- b procedures/subroutines
- c class interfaces

and give examples of their use

5 explain the need for consistency and traceability when naming variables, files, functions, procedures/subroutines, objects, attributes/properties and methods

6 list the information that should be contained in a software specification, see Appendix A.

Unit 301 Develop designs and test software components

Outcome 2 Use diagrammatical representation for design

Practical skills

The candidate will be able to:

1 interpret
   - a an Event/Action chart
   - b a structure chart
   - c an object model
   - d a State Transition Diagram (STD)
   - e an Event-state table
   - f a decision table
   - g process specification eg structured English/pseudocode

2 interpret file, screen and print layouts.

Underpinning knowledge

The candidate will be able to:

1 explain how an Event/Action chart is used to represent the links between events and code for an event-driven program

2 explain how event-driven programs respond to events initiated by a user

3 explain how a structure chart is used to represent the links between modules

4 explain why procedural software modules should have high cohesion and low coupling

5 explain how an object model is used to show the links between objects for an object oriented program

6 describe how objects encapsulate data

7 explain how objects are accessed through their interfaces

8 explain that real-time programs respond to events which cause a change from one state to another

9 explain that a State Transition Diagram provides a diagrammatical representation of the possible states, transitions and enabling events in a real-time program

10 identify that a transition between states takes place as a result of an event occurring

11 explain that each entry in an Event-state table for a real-time program defines
   - a any condition that must be satisfied
   - b the necessary action to be performed as a result of the event
   - c the new state.

Unit 301 Develop designs and test software components

Outcome 3 Specify data types and data structures

Practical skills

The candidate will be able to:

1 document the designs of a program by specifying:

- a variable names and data types
- b argument names and data types
- c return value data types
- d objects, attributes/properties and methods
- e data structures
- f external and internal filename(s)
- g file access method
- h user-defined data types.

Underpinning knowledge

The candidate will be able to:

1 identify the basic data types: integer, floating point, character, string, boolean

2 explain that an object is made up of attributes/properties and methods

3 describe one-dimensional arrays, multi-dimensional arrays and the use of subscripts

4 explain the meaning of the terms file, record and field and describe the relationships between them

5 explain that fields in a file should be as small as possible to reduce the overall file size

6 describe the access methods for files

- a sequential
- b random
- c stream manipulation

7 explain the advantages of using random access files

8 explain the difference between an external filename and an internal filename

9 explain that all accesses to a file must be checked for successful completion

10 list the types of errors that can occur when accessing a file (eg file does not exist, record does not exist, read past end of file, hardware fault)

11 explain the purpose of a user-defined data type

12 explain how a data structure can be used to create the following:

- a table
- b stack
- c queue

d tree

e linked list

13 describe algorithms for sorting:

a bubble sort

b quicksort

14 describe algorithms for searching:

a serial search

b binary search.

Unit 301 Develop designs and test software components

Outcome 4 Develop a software design specification

Practical skills

The candidate will be able to:

1 identify from an outline design specification the software components required

2 select and produce suitable diagrammatical representations for different programming requirements:

- a event-driven
- b procedural
- c object-oriented
- d real-time

3 produce program design language algorithms for components

4 insert references to any pre-written library routines or objects to be used

5 specify

- a the peripherals required
- b error handling routines to trap errors
- c screen error messages
- d security measures
- e constraints (eg file size, hardware)

6 apply validation checks to input data

7 produce file, screen and print layouts

8 check the design for consistency and completeness

9 check that quality criteria is met by the design

10 verify that the design conforms fully with the given specification.

Underpinning knowledge

The candidate will be able to:

1 explain that pseudocode, structured English or flowcharts may be used as program design language

2 explain the advantages of using pre-written library routines or objects (eg built-in, in-house, third-party)

3 explain the need for validation checks on data input

4 identify the different types of validation check that may be performed (eg: range, type (alphabetic, numeric), check digits, hash totals)

5 explain the importance of designing programs to handle error conditions

6 list examples of error conditions that can occur in software

7 explain the need for security measures:

a backup

b passwords

8 explain the benefits of planned incremental implementation and testing of software components

9 describe the quality criteria which should be met by the design:

a effectiveness

b modifiability

c portability

d use of standard routines

e reliability

f reusability

g store usage

h security

i ease of operation.

Unit 301 Develop designs and test software components

Outcome 5 Plan for software testing

Practical skills

The candidate will be able to:

1 produce a plan for carrying out the testing and analyses of results using PERT or Gantt charts

2 identify from a given specification the tests required to carry out testing

3 identify the hardware and software requirements for the system to be tested

4 identify the steps needed to configure the system to be tested and preserve system and data integrity

5 prepare a test plan

6 prepare test data.

Underpinning knowledge

The candidate will be able to:

1 describe the advantages and disadvantages of PERT and Gantt charts for planning

2 describe different types of system that may be tested (eg data processing, data communications, real-time, control, database)

3 explain the different types of test that may be applied

   a functional tests that will ascertain if all components of the system operate correctly with the test data

   b recovery tests that will ascertain if system and data integrity can be maintained under simulated hardware or software failures

   c performance tests that will ascertain that a system can deal with large volumes of data and still achieve the response times required by the user

4 describe how performance testing may require software to be written to generate large volumes of data

5 describe precautions to be taken in applying a test in order for system and data integrity to be preserved (eg backups, informing users, closing down active programs)

6 describe the essential features of a test plan (Appendix B)

   a test number

   b date

   c purpose/type of test

   d expected outputs for stated inputs

7 describe the importance of designing test data to confirm a program works correctly under normal and exceptional circumstances

   a valid

   b invalid

   c boundary

8 state that to comply with quality control procedures a standardised and rigorous approach to testing is required.

Unit 301 Develop designs and test software components

Outcome 6 Record and analyse test results

Practical skills

The candidate will be able to:

1 carry out testing to meet agreed plans

2 use a test plan to carry out a series of tests

3 record the results in a test log

4 provide evidence of testing eg printed output, screen shots, file output

5 carry out analyses of testing to meet agreed plans

6 use the test log to produce a report which:

- a specifies the presence or absence of errors
- b makes proposals for rectifying errors
- c evaluates the test results against the original software specification.

Underpinning knowledge

The candidate will be able to:

1 describe the essential features of a test log (Appendix C)

- a test number
- b date
- c actual results
- d record of discrepancies between actual results and expected results

2 explain how the test number provides a cross reference between a test plan its corresponding test log and test output (printed, screen shots or file)

3 describe the importance of testing software in the target environment

4 explain the need to run a particular test more than once to confirm that a system is working correctly

5 describe the purpose of test reports (eg traceability, consistency, reliability, quality assurance)

6 describe the different types of software error

- a syntax
- b logical
- c run-time
- d non compliance with specification

7 describe common causes of run-time errors eg forever loops, illegal file operations, divide by zero

8 describe the purpose of test plans, test logs, test results and test reports in relation to technical documentation and the relevance of testing in relation to software quality and maintenance

9 describe how well thought out test plans and test data can be reused for subsequent testing after errors are resolved or maintenance amendments made

10 explain the purpose of version control procedures when developing, testing, amending and maintaining software and documentation with reference to quality assurance.

Unit 301 Develop designs and test software components

Outcome 7 Identify Health and Safety requirements

Practical skills

The candidate will be able to:

1 maintain a safe working environment for self and others

2 monitor compliance with relevant parts of Health and Safety procedures

3 use safe working practices at all times

4 operate equipment according to suppliers, manufacturers and/or workplace requirements

5 use and maintain equipment, materials and accessories to a safe standard

6 use reporting procedures to report any hazards

7 carry out formal Health and Safety risk assessment

8 provide guidance to colleagues on Health and Safety.

Underpinning knowledge

The candidate will be able to:

1 describe what elements and practices create a good working environment

- a frequent breaks away from the computer
- b correct positioning of screens/chairs/keyboards
- c adequate lighting and ventilation

2 identify the health and safety precautions to adopt when using a computer

- a ensuring that power cables are safely secured
- b ensuring that power points are not overloaded

3 identify injuries common in a bad working environment

- a repetitive strain injury
- b eye strain
- c bad posture

4 identify cleaning procedures related to IT equipment

5 explain the term ‘ergonomics’

6 describe the main points of relevant legislation: Health and Safety at work Act 1974, electrical regulations, working with VDUs, COSHH regulations

7 explain the importance of keeping fire doors and exits clear and unblocked

8 describe the use of different types of fire extinguishers, in particular those suitable for use in the IT environment.

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 **Explain the concepts of software design**	☐	
2 **Use diagrammatical representation for design**	☐	
3 **Specify data types and data structures**	☐	
4 **Develop a software design specification**	☐	
5 **Plan for software testing**	☐	
6 **Record and analyse test results**	☐	
7 **Identify Health and Safety requirements**	☐	

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

City & Guilds Registration Number ....

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

Appendix A - Information in a software specification

In order to provide a complete basis for design, a program specification will need to contain most or all of the following information:

- **User needs, expectations and experience**
- **Input and output specifications**
  - file formats
  - file access methods
  - record design
  - data definitions
  - data types
  - data structures
  - quantity
  - acceptable values
- **Processing requirements description to be read by:**
  - users
  - management
  - audit
- **Reports required**
  - print layout
  - screen layout
  - file format
- **Code structures to be used**
- **Peripherals to be used**
- **Controls necessary to prove that:**
  - correct files were input
  - all of the data input has been processed
  - all of the data is accounted for on output
- **Quality criteria to be met by the design, particularly in terms of:**
  - effectiveness
  - modifiability
  - portability
  - use of standard routines
  - store usage
  - security
  - ease of operation
  - reliability
  - re-usability
- **Other criteria**
  - run-time parameters
  - operator-controlled insertions (if any)
  - system-generated data
  - error reports
  - management information
- **Program test plan**
  - data in
  - expected output

Appendix B

1 TEST PLAN

SYSTEM NAME: PROGRAM NAME: MODULE NAME:			VERSION NO: TESTER NAME:	PAGE NO:
Test No	**Date**	**Purpose/Type Of Test**	**Input/Filename**	**Expected Output/Filename**

2 Appendix C

3 TEST LOG

SYSTEM NAME: PROGRAM NAME: MODULE NAME:			VERSION NO: TESTER NAME:	PAGE NO:
Test No	**Date**	**Actual Output/Filename**	**Comments on discrepancies**	

Published by City & Guilds
1 Giltspur Street
London
EC1A 9DD
T +44 (0)20 7294 2468
F +44 (0)20 7294 2400
www.cityandguilds.com
www.cityandguilds.com/e-quals07

City & Guilds is a registered charity established to promote education and training