GCE

Rewarding Learning

CCEA GCE Specification in

Software Systems Development

Updated: September 2019

For first teaching from September 2016
For first award of AS level in Summer 2017
For first award of A level in Summer 2018
Subject Code: CL4

Foreword

This booklet contains CCEA's Advanced Subsidiary (AS) and Advanced GCE Software Systems Development for first teaching from September 2016.

The AS is the first part of the full Advanced GCE course. It is possible to take the AS as a stand-alone qualification. The AS units are assessed at a standard appropriate for students who have completed half of the full course.

The A2 is the second part of the full Advanced GCE course. Assessed at a standard appropriate for students who have completed the full course, the A2 units include both synoptic assessment (to assess students' overall learning throughout the course) and an element of stretch and challenge.

The full Advanced GCE award is based on students' marks from the AS (40 percent) and the A2 (60 percent). We award an A* to students who achieve a grade A in the full A Level qualification and at least 90 percent of the maximum uniform marks available across the A2 units.

We will notify centres in writing of any major changes to this specification. We will also publish changes on our website at www.ccea.org.uk

The version on our website is the most up-to-date version. Please note that the web version may be different from printed versions.

Updated: September 2019

Subject Code	CL4
QAN AS	601/8913/7
QAN A2	601/8914/9

You may download further copies of this publication from www.ccea.org.uk

Updated: September 2019

Contents

Updated: September 2019

1 Introduction

This specification sets out the content and assessment details for our Advanced Subsidiary (AS) and Advanced Level (A Level) courses in Software Systems Development. First teaching begins from September 2016. You can view and download the latest version of this specification on our website at www.ccea.org.uk

Students can take the AS course as a final qualification or as the first half of the A Level course. They must also complete the A2 course (the second half of the A Level) if they wish to obtain a full A Level qualification. We will make the first AS awards for this specification in 2017 and the first A Level awards in 2018.

The specification builds on the broad objectives of the Northern Ireland Curriculum.

Our GCE in Software Systems Development is an applied qualification in which students develop knowledge, understanding and skills through practical application in a context related to employability.

As with all GCEs, the guided learning hours for this specification are:

- 180 hours for the Advanced Subsidiary award; and
- 360 hours for the Advanced Level award.

1.1 Aims

This specification aims to encourage students to:

- develop a genuine interest in software systems development with a focus on programming;
- develop an understanding of systems approaches and modelling techniques to support software development;
- develop software development skills that will prepare them for work in today's software industry;
- participate in the development of a software project using a complete software development process;
- apply their skills to relevant work-related scenarios;
- work with others in group settings;
- research, develop and present their findings in a variety of formats; develop advanced study skills in preparation for third level education; and demonstrate their understanding and application of key concepts through challenging internal and external assessment.

1.2 Key features

The key features of the specification appear below.

- It includes four assessment units: two externally assessed and two internally assessed.
- The content directly relates to current software development trends supported by systems methodologies.
- The content is well balanced between knowledge and application.
- The specification will develop real practical skills in an object oriented environment.
- Assessment at A2 includes stretch and challenge, reflected in the use of a wider range of question types, higher demand evaluative tasks, synoptic assessment and extended writing.
- A course of study based on this specification provides a sound basis for progression to higher education.

1.3 Prior attainment

There is no particular level of attainment required to study this specification. However, it builds on some of the knowledge, understanding and skills developed in GCSE ICT.

1.4 Classification codes and subject combinations

Every specification is assigned a national classification code that indicates the subject area to which it belongs. The classification code for this qualification is CL4.

Progression to another school/college

Should a student take two qualifications with the same classification code, schools and colleges that they apply to may take the view that they have achieved only one of the two GCEs. The same view may be taken if students take two GCE qualifications that have different classification codes but have content that overlaps significantly. Students who have any doubts about their subject combinations should check with the universities and colleges that they wish to attend before embarking on their planned study.

2 Specification at a Glance

The table below summarises the structure of the AS and A Level courses:

Content	Assessment	Weightings	Availability
AS 1: **Introduction to Object Oriented Development**	External written examination **2 hours** Short and extended questions, stimulus response and data response questions based on the principles of object oriented development	50% of AS 20% of A Level	Every Summer From 2017
AS 2: **Event Driven Programming**	Internal assessment Portfolio showing evidence of designing, implementing, testing and evaluating an event driven application	50% of AS 20% of A Level	Every Summer From 2017
A2 1: **Systems Approaches and Database Concepts**	External written examination **2 hours** Short and extended questions relating to current systems approaches and database concepts These questions are based on a pre-release case study.	30% of A Level	Every Summer From 2018
A2 2: **Implementing Solutions**	Internal assessment Portfolio showing evidence of the analysis, design and implementation of a software solution of a specified problem in a pre-release case study and task.	30% of A Level	Every Summer From 2018

3 Subject Content

We have divided the course into four units: two units at AS level and two units at A2. The content of each unit and the respective learning outcomes appear below. Students will develop significant programming skills in the language of their choice. In the final A2 unit, students will build a Relational Database Management Systems (RDMS) application through an object oriented environment.

3.1 Unit AS 1: Introduction to Object Oriented Development

This unit provides students with a thorough understanding of object oriented systems. Students adopt an object oriented approach to problem solving. Object concepts are defined and implemented. The unit enables students to develop object oriented skills. It helps students to appreciate the benefits of developing applications in this type of environment. This unit is externally assessed through a two hour question paper.

Content	**Learning Outcomes**
Software	Students should be able to: • demonstrate knowledge and understanding of the concepts associated with: – systems software (management of file types/source/binary/executable/XML); and – application development software (object oriented platforms including open source);
Defining Data	• explain and apply: – primitive type; and – reference (object) types; • identify and name objects; • explain that an object is an occurrence of a class; • identify and name classes (abstract, base and inherited classes); • examine and apply the functions of classes (reusability); • explain the use of a static variable within a class; • examine and apply attributes and types to a class; • identify that attributes represent the properties of a class; and • identify the use of interfaces in relation to specific classes.

Content	Learning Outcomes
Program Control Structures	Student should be able to: • apply and evaluate the basic principles of control structures in terms of: – sequence (sequential functions, methods); – repetition (unconditional, conditional); and – selection (decision IF, nested IF, switch);
Objects	• apply and discuss the principles underpinning encapsulation, inheritance and polymorphism, for example: – visibility (public, private and protected); and – overloading and overriding methods;
Data Structures	• explain the need to store and organise data efficiently within specific structures; • explain, design and use the following data structures: – strings; and – static arrays (of simple type and of objects);
Exception Handling	• explain the need to be able to trap errors in program code; • explain ways that errors can be trapped in an object oriented environment and apply associated techniques; • try/catch (blocks);
Managing Input/Output	• input data from the command line prompt; • control screen output; • evaluate the role of different file types in an object oriented environment including: – text files; – binary files; and – object files (serialisation);
Testing an Object Oriented Application	• assess the need to apply basic tests to object oriented applications; • develop and use suitable test plans; • evaluate test plans; and • analyse, evaluate and interpret test results.

3.2 Unit AS 2: Event Driven Programming

This unit provides students with an opportunity to implement and develop object oriented technologies in an event driven environment. Students are able to state requirements and design, implement, test and evaluate their application. This unit is internally assessed. We provide further guidance on internal assessment in Section 6.

Content	Learning Outcomes
Defining Graphical User Interface (GUI) Objects	Students should be able to: • demonstrate their understanding and use of GUI objects in the implementation of an event driven application; • understand and apply the features of: – GUI; and – forms (common components);
Understanding Events	• demonstrate and apply their understanding of events in the implementation of an event driven application; • use triggers, for example button, mouse clicks or key presses;
Using Multiple Forms	• understand the use and application of multiple forms in an event driven application; • design and implement multiple forms for an event driven application; • implement appropriate navigation: – menus; – toolbars; and – buttons;
Designing an Event Driven Application	• illustrate the concept of a storyboard and its use in an event driven application; • develop storyboards for event driven applications with respect to user requirements and functionality; and
Linking an Object Application to Simple Files	• implement: – text files; – binary files; and – object files (object serialisation).

Content	Learning Outcomes
Testing an Event Driven Application **Evaluating an Event Driven Application**	Students should be able to: • apply the concepts of: – error trapping techniques from an object perspective; and – try/catch (blocks); and • design and apply a suitable test strategy to an event driven application. • document, interpret and evaluate test results; and • evaluate an event driven application in terms of: – requirements; – features; and – functionality.

3.3 Unit A2 1: Systems Approaches and Database Concepts

This unit provides students with a thorough understanding of the reasons for systems development. It also provides them with an understanding of fundamental systems analysis and design concepts. It provides a detailed study of design methodologies. The unit introduces students to project management concepts and testing strategies that assist the systems development process.

The unit introduces important database concepts enabling the student to understand relational database systems, implemented through Structured Query Language (SQL). This unit is externally assessed through a two hour question paper with a pre-release case study. CCEA will publish the A2 1 pre-release case study in March each year for use in that examination series.

Content	Learning Outcomes
Reasons for Systems Development	Students should be able to: • analyse the reasons for project initiation and systems development within organisations; • examine and apply analysis and design principles to business solutions; • justify the roles inherent in the systems development process, particularly the role of the user;
Systems Methodologies	• explain the purpose of applying a methodology to the solution of systems problems; • discuss how methodologies have evolved in line with changing technologies and discuss the roles of participants in their application; • assess the application of a range of methodologies including traditional, iterative and Agile methods: – Waterfall; – RAD; – DSDM; – SCRUM; and – XP; • examine the application of methodologies in specific situations; • explain stages, rules, tools and techniques associated with methodologies; and • examine the role of prototyping in the development process.

Content	**Learning Outcomes**
Systems Methodologies (cont.)	Students should be able to: • explain the basic concepts of UML and apply some simple modelling techniques;
Managing Projects	• discuss the purpose of project management; • examine and apply project management concepts; • examine the roles and responsibilities of those working in project management; • explain the resources involved in developing a project; • apply project planning techniques including the use of Gantt, Pert, Milestones, Float and Critical Path;
Testing	• assess the significance of testing throughout the development cycle, particularly at the design stage; • discuss the different types of testing that can be applied throughout the development cycle including program, system, white box, black box, integration and user acceptance, alpha and beta; • explain the role of the personnel involved in testing procedures; • examine the concept of a test plan and its likely content; • develop and apply test plans for specific scenarios;
Database Concepts	• examine the concept of a database and associated terminology; • explain the nature of databases: – relational models; – hierarchical models; and – network;
Entity Relationship (ER) Models	• explain ER models: – entities; and – relationships; and • identify and explain different types of relationships: – one to one; – one to many; and – many to many.

Content	Learning Outcomes
Normalisation	Students should be able to: • analyse and apply normalisation including: – first normal form; – second normal form; and – third normal form; and
Structured Query Language (SQL)	• demonstrate implementing data using SQL commands and statements by applying: – principles of SQL; – SQL commands; and – SQL statements.

3.4 Unit A2 2: Implementing Solutions

This unit provides students with an opportunity to design and implement a solution to a given problem using the knowledge and skills acquired in the preceding units. The students implement an agreed design using an appropriate software tool. The unit allows them to experience the elements of the systems development process. We require students to build their solutions using an RDMS through an event driven programming environment. This unit is internally assessed with a pre-release case study. Students must use the pre-release case study throughout. We provide further guidance on internal assessment in Section 6.

Content	Learning Outcomes
Applying Project Management Techniques to the Development Process	Students should be able to: • create a detailed project plan for the development process that is capable of reflecting change as the project progresses; • document the plan using relevant tools and techniques; • update the plan to reflect change;
Selecting a Systems Approach to the Solution of a Business Problem	• evaluate different approaches to a given problem; • select and justify a specific approach to a given problem;

Defining and Documenting User Requirements	• present and explain a detailed user requirements specification for an identified problem, considering:
	– functional requirements;
	– non-functional requirements;
	– software; and
	– hardware;

Documenting the Design of the Solution and Testing the Design	• provide a detailed design of their solution including: – a suitable data model (for example an ER diagram); and – a description of the inputs, outputs, processes, tables, screen designs and reports; and • test the design of their solution to ensure that the desired user outcomes can be achieved.

Content	**Learning Outcomes**
Developing and Implementing a Desktop Solution Using an RDMS through an Event Driven Programming Environment	Students should be able to: • apply techniques of relational database management systems to solutions to the given problem; • use an appropriate software tool to connect the required database to the GUI for the specified problem; • implement the solution using: – multiple forms; – appropriate navigation; and – connection to the database through data adaptors, data sets and data commands; • present the solution using appropriate media;
Testing the Solution	• apply suitable testing procedures to their software solution; • document test outcomes; • evaluate their test procedures;
Evaluating the Solution	• conduct a critical evaluation of their solution that refers to the user requirements specification and the project plan; and • provide an evaluation of their personal performance.

4 Scheme of Assessment

4.1 Assessment opportunities

Each unit is available for assessment in summer each year. It is possible to resit individual AS and A2 assessment units once and count the better result for each unit towards an AS and A level qualification. Candidates' results for individual assessment units can count towards a qualification until we withdraw the specification.

4.2 Assessment objectives

Below are the assessment objectives for this specification. Candidates must:

- demonstrate knowledge and understanding of concepts, systems approaches, and solutions relevant to software systems development (AO1);
- apply their knowledge and understanding to develop and implement solutions to the problems identified (AO2); and
- analyse and evaluate the concepts of software systems development and candidates' own performance in problem solving (AO3).

4.3 Assessment objective weightings

The table below sets out the assessment objective weightings for each assessment unit and the overall A Level qualification:

Percentage Assessment Objective Weightings					
	AO1	AO2	AO3	AS	A level
AS 1	7.2%	7.2%	5.6%	20%	20%
AS 2	5.6%	8.8%	5.6%	20%	20%
A2 1	9.6%	10.8%	9.6%		30%
A2 2	6%	14.4%	9.6%		30%
Total	**28.4%**	**41.2%**	**30.4%**	**40%**	**100%**

4.4 Quality of written communication

In AS and A Level Software Systems Development, candidates must demonstrate their quality of written communication (QWC). In particular, they must:

- ensure that text is legible and that spelling, punctuation and grammar are accurate so that meaning is clear;
- select and use a form and style of writing appropriate to their purpose and to complex subject matter; and
- organise information clearly and coherently, using specialist vocabulary where appropriate.

Examiners and teachers assess the quality of candidates' written communication in their responses to questions and tasks that require extended writing.

4.5 Synoptic assessment at A2

The A2 assessment units include an element of synoptic assessment, which encourages the development of the understanding of the subject as a whole. In Software Systems Development, synoptic assessment requires candidates to demonstrate that they can:

- build on material first encountered in the AS units;
- draw together the knowledge, understanding and skills learned in different parts of the A Level course;
- select and present work for examination that demonstrates their strengths across the areas of knowledge and the range of skills described;
- sustain their own lines of enquiry and record and observe from primary sources;
- bring together and make connections between the areas of knowledge and the range of skills described and learned throughout the course; and
- respond to one or more of the following:
  - a stimulus or issue;
  - a design brief or problem; and/or
  - a task that specifies an image, object or other outcome to be achieved.

4.6 Stretch and challenge at A2

The A2 assessment units provide opportunities for stretch and challenge by incorporating:

- a wide range of question types to address different skills, for example case studies and open-ended questions;
- high level of demand evaluative tasks;
- questions that require candidates to show more connections between sections of the specification; and
- extended writing where appropriate.

4.7 Reporting and grading

We report the results of individual assessment units on a uniform mark scale that reflects the assessment weighting of each unit.

We award AS qualifications on a five grade scale from A to E, with A being the highest. We award A Level qualifications on a six grade scale from A* to E, with A* being the highest. We determine the AS and A Level grades awarded by aggregating the uniform marks obtained on individual assessment units. To be awarded an A*, candidates need to achieve a grade A on their full A Level qualification and at least 90 percent of the maximum uniform marks across the A2 units. If candidates fail to attain a grade E, we report their results as unclassified (U).

The grades we award match the performance descriptions in Section 5 of this specification.

5 Grade Descriptions

Grade descriptions are provided to give a general indication of the standards of achievement likely to have been shown by candidates awarded particular grades. The descriptions must be interpreted in relation to the content in the specification; they are not designed to define that content. The grade awarded depends in practice upon the extent to which the candidate has met the assessment objectives overall. Shortcomings in some aspects of candidates' performance in the assessment may be balanced by better performances in others.

AS Grade Descriptions

Grade	Description
AS **Grade A**	For **AO1**, candidates characteristically: • demonstrate thorough knowledge and understanding of the concepts and underpinning principles of software systems development from an object oriented perspective; • demonstrate ability to design and determine the need to test event driven applications using appropriate methods; and • use accurate, appropriate technical language associated with software systems development. For **AO2**, candidates characteristically: • effectively apply knowledge and understanding of the concepts and underpinning principles of software systems development from an object oriented perspective to the solution of complex problems in an object oriented environment; • apply object oriented approaches to problem solving and effectively apply testing techniques; and • articulate solutions to problems using technical language appropriately and accurately. For **AO3**, candidates characteristically: • appreciate the value of developing applications in an object oriented environment; • discuss the concepts and underpinning principles of software systems development from an object oriented perspective; • comprehensively justify strategies, articulate and evaluate solutions to problems in an object oriented environment; and • critically analyse and evaluate test strategies and test results.

Grade	Description
AS **Grade E**	For **AO1**, candidates characteristically: • demonstrate basic, sometimes incomplete knowledge and understanding of the concepts and underpinning principles of software systems development from an object oriented perspective; • demonstrate limited ability to design and understand the need to test event driven applications using appropriate methods; and • use basic levels of technical language associated with software systems development. For **AO2**, candidates characteristically: • apply basic knowledge and understanding of the concepts and underpinning principles of software systems development from an object oriented perspective to the solution of simple problems in an object oriented environment; • apply basic object oriented approaches to problem solving and limited application of testing techniques; and • suggest solutions to problems using basic technical language. For **AO3**, candidates characteristically: • demonstrate basic understanding of the value of developing applications in an object oriented environment; • provide limited discussion of the concepts and underpinning principles of software systems development from an object oriented perspective; • provide basic justification of strategies and ability to articulate and evaluate solutions to problems in an object oriented environment; and • demonstrate limited analysis and evaluation of test strategies and test results.

A2 Grade Descriptions

Grade	Description
A2 **Grade A**	For **AO1**, candidates characteristically: • demonstrate advanced knowledge and understanding of the concepts of the systems development process and in the solution of advanced problems; • define user requirements using accurate, appropriate technical language; and • demonstrate detailed knowledge of project management, testing procedures and database concepts in the systems development process. For **AO2**, candidates characteristically: • apply advanced knowledge and understanding of the concepts of the systems development process and analytical approaches to problem solving; • document user requirements using accurate, appropriate technical language; and • apply appropriate project management techniques in implementing and testing a desktop solution using an RDMS though an event driven programming environment. For **AO3**, candidates characteristically: • comprehensively justify strategies for solving a given problem; • comprehensively analyse and evaluate test strategies and test results/outcomes drawing valid conclusions; and • critically evaluate the solution with regard to user requirements and the project plan and critically evaluate personal performance.

Grade	Description
A2 **Grade E**	For **AO1**, candidates characteristically: • demonstrate basic knowledge and understanding of the concepts and a sometimes incomplete knowledge of systems approaches in the solution of advanced problems; • define user requirements using limited technical language; and • demonstrate basic, sometimes incomplete knowledge of project management testing procedures and database concepts in the systems development process. For **AO2**, candidates characteristically: • apply basic knowledge and understanding of the concepts of the systems

Grade	Description
	development process and basic approaches to the solution of a problem; • document user requirements using basic technical language; and • apply limited project management techniques in implementing and testing a basic desktop solution using an RDMS though an event driven programming environment. For **AO3**, candidates characteristically: • provide a basic justification of strategies for solving a given problem; • provide a basic analysis and evaluation of test strategies used and test results obtained; • provide test outcomes drawing simple and sometimes incomplete conclusions; and • provide a basic evaluation of the solution, making limited reference to user requirements and the project plan and basic evaluation of personal performance.

6 Guidance on Internal Assessment

There are two internal assessments in this specification, one at AS Level and one at A2:

- Internal Assessment Unit AS 2: Event Driven Programming; and
- Internal Assessment Unit A2 2: Implementing Solutions.

These units each represent approximately 60 hours of work. The internal assessment will have a specific focus on candidates' ability to apply their knowledge and skills within a work-related context.

6.1 Setting of tasks

We will provide centres with details of the assessment tasks and guidance on how to complete and submit them. Teachers must ensure that the completed tasks conform to the unit requirements. We will change the A2 2 pre-release case study and task every year to ensure that it continues to set an appropriate challenge and remains valid, reliable and stimulating. We will publish the pre-release case study and task for A2 2 in June each year for use the following year.

6.2 Supervision of candidates

The teacher should ensure that the candidate completes the internal assessment in line with the specification requirements. The teacher should also ensure that the assessment can be marked using the criteria set out for each unit.

The teacher should supervise the work at all times to:

- monitor the candidate's progress;
- prevent plagiarism and repetition; and
- advise them on the most concise and effective methods to use when presenting their work.

Teachers must sign a declaration to certify that, to the best of their knowledge, all the work the candidate has submitted for assessment is their own. Teachers must be mindful of the fact that it is possible to download code development software from the internet.

For up-to-date advice on plagiarism, or any kind of candidate malpractice, teachers should refer to the Joint Council for Qualifications' *Suspected Malpractice in Examinations and Assessments: Policies and Procedures* on the JCQ website: www.jcq.org.uk

6.3 Assessment conditions

Candidates will complete their internal assessment in conditions that combine classroom and independent study activities. These conditions must ensure that the internal assessment remains reliable and fair.

6.4 Marking and internal standardisation

Teachers should use their professional judgement to select and apply the criteria in each mark band appropriately and fairly to candidates' work. Within the mark band teachers should use a 'best fit' approach as to whether the response fulfils performance descriptors and may be described as basic/satisfactory/good/excellent in terms of performance.

Centres with more than one teaching group must carry out internal standardisation of the internal assessment tasks before submitting them to us. This is to ensure, as far as possible, that each teacher has applied the assessment criteria consistently when marking assessments.

As a result of internal standardisation, it may be necessary to adjust an individual teacher's marking. This is to bring assessments into line with those of other teachers in the centre and to match the standards established at the agreement trial. Where adjustment is necessary, the total/final mark recorded on the Candidate Record Sheet should be amended.

6.5 Moderation

Centres must submit their marks and samples to us by a specified date in May in any year. We may adjust centres' marking. This is to bring the assessment of the candidates' work into line with our agreed standards. We issue full instructions well in advance of submission on:

- the details of moderation procedures;
- the nature of sampling; and
- the dates by which marks and samples must be submitted to us.

Teachers and centre staff may contact us at any stage if they require advice, assistance or support regarding any aspect of internal assessment. We provide moderators who can support groups of centres or contact individual centres to discuss issues arising from the internal assessments.

7 Links

7.1 Support

We provide the following resources to support this specification:

- a subject microsite within our website;
- guidance notes for teachers; and
- specimen assessment materials.

We intend to expand our range of support to include the following:

- past papers and mark schemes;
- Chief Examiner's reports;
- Principal Moderator's reports;
- schemes of work;
- centre support visits;
- support days for teachers;
- portfolio clinics;
- agreement trials;
- a resource list; and
- exemplification of standards.

7.2 Curriculum objectives

This specification addresses and builds upon the broad curriculum objectives for Northern Ireland. In particular, it enables students to understand, relate to and explore:

- sustainable development (addressed in AS 1: Introduction to Object Oriented Development and AS 2: Event Driven Programming);
- health and safety considerations (addressed particularly in the practical units AS 2: Event Driven Programming and A2 2: Implementing Solutions); and
- employability and 'skills agenda' in developing software systems development skills which are transferable across Europe (addressed throughout the specification).

7.3 Skills development

This specification provides opportunities for students to develop the following key skills:

- information and communications technology – for example event driven programming (AS 2) and implementing solutions (A2 2);
- communication – assessed through the quality of written communication, for example in externally assessed units (AS 1 and A2 1); and
- application of number – for example problem solving (AS1) and project management (A2 1 and A2 2).

7.4 Examination entries

Entry codes for this subject and details on how to make entries are available on our Qualifications Administration Handbook microsite, which you can access at www.ccea.org.uk

Alternatively, you can telephone our Examination Entries, Results and Certification team using the contact details provided in this section.

7.5 Equality and inclusion

We have considered the requirements of equality legislation in developing this specification.

GCE qualifications often require the assessment of a broad range of competences. This is because they are general qualifications and, as such, prepare students for a wide range of occupations and higher level courses.

During the development process, an external equality panel reviewed the specification to identify any potential barriers to equality and inclusion. Where appropriate, we have considered measures to support access and mitigate barriers.

Reasonable adjustments are made for students with disabilities in order to reduce barriers to accessing assessments. For this reason, very few students will have a complete barrier to any part of the assessment. However, students with hearing impairments may be restricted in the type of media they use for the internal assessment, for example aspects of audio visual media.

It is important to note that where access arrangements are permitted (including the use of British Sign Language/Irish Sign Language Interpreters) they must not be used in any way that undermines the integrity of the assessment. You can find information on reasonable adjustments in the Joint Council for Qualifications' document *Access Arrangements and Reasonable Adjustments: General and Vocational Qualifications*, available at www.jcq.org.uk

7.6 Contact details

The following list provides contact details for relevant staff members and departments:

- Specification Support Officer: Nuala Tierney
  (telephone: (028) 9026 1200, extension 2292, email: ntierney@ccea.org.uk)

- Officer with Subject Responsibility: Andrew Douglas
  (telephone: (028) 9026 1200, extension 2713, email: adouglas@ccea.org.uk)

- Examination Entries, Results and Certification
  (telephone: (028) 9026 1262, email: entriesandresults@ccea.org.uk)

- Examiner Recruitment
  (telephone: (028) 9026 1243, email: appointments@ccea.org.uk)

- Distribution
  (telephone: (028) 9026 1242, email: distribution@ccea.org.uk)

- Support Events Administration
  (telephone: (028) 9026 1401, email: events@ccea.org.uk)

- Information Section (including Freedom of Information requests)
  (telephone: (028) 9026 1200, email: info@ccea.org.uk)

- Business Assurance (Complaints and Appeals)
  (telephone: (028) 9026 1244, email: complaints@ccea.org.uk or appealsmanager@ccea.org.uk)

- Moderation
  (telephone: (028) 9026 1200, extension 2236, email: moderationteam@ccea.org.uk)

Appendix 1

Glossary of terms

Students should be familiar with the following terms and the definitions provided below.

DSDM	Dynamic Systems Development Method
ER	Entity Relationship
GUI	Graphical User Interface
RAD	Rapid Application Development
RDMS	Relational Database Management System
SQL	Structured Query Language
XML	Extensible Markup Language
XP	Extreme Programming