Oxford Cambridge and RSA Examinations

OCR AS GCE in Computing (3820)

OCR Advanced GCE in Computing (7820)

Approved Specifications – Revised Edition

First Advanced Subsidiary GCE certification was 2001	QAN (3820) 100/0080/X
First Advanced GCE certification was 2002	QAN (7820) 100/0079/3

Foreword to Revised Edition

This Revised Edition has been produced to consolidate earlier revisions to these specifications and any changes contained within have previously been detailed in notices to centres**. There is no change to the structure or teaching content of the specification and most differences are cosmetic**. Sidelining will be used to indicate any significant changes.

The main changes are:

Re-sits of Units – The restrictions on re-sitting units have been removed, enabling candidates to re-take units more than once (for details see page 16).

Synoptic Assessment - It is no longer a requirement to take synoptic units at the end of the course (for details see page 15).

Foreword (continued)

This booklet contains OCR's Advanced Subsidiary GCE (AS) and Advanced GCE (A level) Computing specifications for teaching from September 2004.

The AS GCE is assessed at a standard appropriate for candidates who have completed the first year of study of a two year Advanced GCE course, i.e. between GCSE and Advanced GCE. It forms the first half of the Advanced GCE course in terms of teaching time and content. When combined with the second half of the Advanced GCE course, known as 'A2', the AS forms 50% of the assessment of the total Advanced GCE. However, the AS can be taken as a 'stand-alone' qualification. A2 is weighted at 50% of the total assessment of the Advanced GCE.

In these specifications the term **module** is used to describe specific teaching and learning requirements. The term **unit** describes a unit of assessment.

Each teaching and learning module is assessed by its associated unit of assessment.

These specifications meet the requirements of the Common Criteria (Qualifications and Curriculum Authority, 1999), the GCE Advanced Subsidiary and Advanced Level Qualification-Specific Criteria (QCA, 1999) and the relevant Subject Criteria (QCA, 1999).

Contents

Specification Summary

Outline

These specifications meet the requirements of the AS (AS) and Advanced GCE Subject Criteria for Computing and the Advanced GCE Common Criteria (QCA, 1999). The first three units of the specification make up the AS, with a further three units required to complete the Advanced GCE.

The overall aim of the specifications is to encourage candidates to develop an understanding of the principles of problem solving using computers and of the range of applications of computers, and the effects of their use, in order to apply this understanding to develop computer-based solutions to problems. Candidates develop an understanding of systems analysis and design, development methods, testing, implementation and documentation.

Rationale

In today's world, individuals will increasingly need technological and computing skills that include the ability to understand problems suitable for solution by computer, to construct such a solution, while being fully aware of its wider implications and possessing the systematic methodology required for the successful delivery of such a solution.

The impact of computers on society is enormous and, as the percentage of businesses and households that utilise computer-based technologies continues to grow, so does the need for individuals who can master and manipulate these new systems. As well as the rapid development of new and ever more powerful processors, the complexity, sophistication and power of software has also increased.

Broad Aims

These specifications are designed to develop an understanding of the fundamentals of computing and to provide the knowledge and skills suitable for participation in an evolving computer-dependent society. Computing provides opportunities to be at the forefront of these changes and for students, through their careers, to play a role in the dynamic computer-based information society of the future.

Because the effects of computer technologies are so far reaching, students must be aware of the ever-growing impacts of computers. They need to reflect critically on the role of the computer in society and to consider the positive and negative effects of its use. These specifications reflect this need to develop skills and attitudes that increase students' abilities to address the social and ethical issues of technological advancements.

Progression opportunities

In today's workplace, those with knowledge and skills in computing have the opportunity to pursue new and exciting careers and to be instrumental in the conception of computer systems that increasingly shape work and leisure activities.

To meet these career challenges, students must be self-reliant as well as good communicators and problem solvers. They require interpersonal, academic, and technical skills, and must demonstrate an ability to work independently and as part of a team. They also need to develop an ethical approach to the use of computers. These specifications provide a focus to develop these skills, while ensuring that students acquire a sound knowledge of computing.

It is envisaged that students will utilise the skills and knowledge of computing in one of three ways. Firstly, to provide a general understanding and perspective of the use of computer technology and systems, which will inform their decisions and support their participation in an increasingly technologically dependent society. Secondly, to provide the necessary skills and knowledge to seek employment in areas that utilises computing, where they may develop their skills and knowledge further through practical experience and training. Thirdly, students may choose to continue to develop their knowledge and understand of computing through entry to higher education, where this qualification will provide a useful foundation for further study of computing or more specialist aspects of computing.

Recommended prior learning

No previous knowledge of Computing is assumed although these specifications develop the aims of GCSE specifications in Information Technology by enhancing and broadening the range and the understanding of the capabilities of applications and information systems studied. The use of a range of tools and techniques is required to solve more extensive and more complex problems.

Specification Content

AS

Module 2506 covers the core aspects of computer systems. Practical skills of design, development, testing and implementation are developed further in Module 2507 with structured practical tasks set by OCR and assessed by individual centres. To put into practice these practical skills it is a pre-requisite that appropriate parts of Module 2508 will be studied, in which the system development life cycle is covered with reference to particular applications.

A2

Module 2509 develops concepts already studied in Module 2506 to a much greater depth and introduces two new concepts: programming paradigms and databases. In Module 2510, candidates develop further their knowledge and understanding of computer systems and the skills covered in Module 2507. The project in Module 2510 requires analysis, design, development, testing and implementation of a real-life problem, which is carried out over an extended period. The whole project is evaluated, organised and presented in a report.

In Module 2511, the emphasis is on the application of theory. The module builds on underlying concepts studied in previous modules and introduces new concepts of project management, simulation and real-time processing and common network environments.

Scheme of Assessment

The AS GCE forms 50% of the assessment weighting of the full Advanced GCE. AS GCE is assessed at a standard between GCSE and Advanced GCE and can be taken as a stand-alone specification or as the first part of the full Advanced GCE course.

Assessment is by means of **three units of assessment** for AS GCE and **six units of assessment** for Advanced GCE.

AS GCE	Candidates take Units 2506, 2507 and 2508.
Advanced GCE	Candidates take Units 2506, 2507, 2508, 2509, 2510 and 2511.

Units of Assessment

Unit	Level	Name	Mode of Assessment	Duration	Weighting	
					AS	Advanced GCE
2506	AS	Introductory Computer Systems, Communications and Software	Written Examination	1 hour 30 mins	30%	15%
2507	AS	Structured Practical Computing Tasks	Coursework	-	40%	20%
2508	AS	Computer Systems Development and Practical Applications	Written Examination	1 hour 30 mins	30%	15%
2509	A2	Systems Software Mechanisms, Machine Architecture, Database Theory and Programming Paradigms	Written Examination	1 hour 30 mins	-	15%
2510	A2	Computing Project	Coursework	-	-	20%
2511	A2	Integrated Information Systems	Written Examination	1 hour 30 mins	-	15%

Question Paper Requirements

AS

Unit 2506: Introductory Computer Systems, Communications and Software: a written paper of 1 hour 30 minutes duration, containing 10-15 compulsory questions requiring short/paragraph length answers. Questions are not based in a context.

Unit 2508: Computer Systems Development and Practical Applications: a written paper of 1 hour 30 minutes duration, containing 6-8 compulsory structured questions based in a variety of contexts.

A2

Unit 2509: Systems Software Mechanisms, Machine Architecture, Database Theory and Programming Paradigms: a written paper of 1 hour 30 minutes duration, containing 6-8 compulsory structured questions, some of which relate to a context.

Unit 2511: Integrated Information Systems: a written paper of 1 hour 30 minutes duration, containing 5-7 compulsory questions set in the context of a short scenario.

Coursework Requirements

AS

Unit 2507 Structured Practical Computing Tasks. Pre-released OCR set tasks are presented to the candidates. These tasks require candidates to find solutions to given problems (including an algorithm) and to demonstrate competence in the skills of Design, Software Development, Testing and Implementation. The candidate’s work is marked by the teacher and externally moderated by OCR.

A2

Unit 2510 Computing Project. Candidates are required to develop a computer-based solution to a real problem. The candidate and the teacher should negotiate to formulate the problem definition. Candidates are required to produce work requiring Analysis, Design, Development, Testing, Implementation, Documentation and Evaluation. The candidate’s work is marked by the teacher and externally moderated by OCR.

Overlap with other qualifications

Overlap with Advanced GNVQ is detailed in Appendix D.

1 Introduction

Computing has facilitated significant changes in society over the past 40 years. These specifications provide an opportunity to develop an understanding of the ability of mankind to advance technologically and hence to improve quality of life.

These specifications are designed to allow AS GCE and Advanced GCE candidates the opportunity to develop their understanding of problem solving using the principles of computing, and to develop a systematic approach to systems analysis and design, methods of problem formulation and production of solutions using computers.

All candidates have the opportunity to do coursework: in AS this is a series of OCR-set structured practical computing tasks; and, in the A2 half of the Advanced GCE, as a substantial piece of coursework – the Computing Project.

The acquisition of basic knowledge and understanding goes hand in hand with its application in the form of the skills of analysis, design, development, testing, implementation, documentation and evaluation.

These specification conform to the QCA Subject Criteria for Computing and to the *Common Criteria for A Level* (QCA, 1999).

1.1 Certification Title

These qualifications are shown on a certificate as:

- OCR Advanced Subsidiary GCE in Computing.
- OCR Advanced GCE in Computing.

1.2 Language

These specifications and associated assessment materials are available in English only.

1.3 Overlap with GNVQ

Overlap with Advanced GNVQ Information and Communications Technology is detailed in Appendix D.

1.4 Exclusions

Candidates who enter for this AS GCE specification may **not** also enter for any other AS GCE specification with the certification title Computing, Information Technology or Information and Communications Technology, in the same examination session.

Candidates who enter for this Advanced GCE specification may **not** also enter for any other Advanced GCE specification with the certification title Computing, Information Technology or Information and Communications Technology, in the same examination session.

Every specification is assigned to a national classification code indicating the subject area to which it belongs.

Centres should be aware that candidates who enter for more than one GCE qualification with the same classification code will have only one grade (the highest) counted for the purpose of the School and College Performance Tables.

The classification code for these specifications is 2610.

1.5 Code of Practice Requirements

These specifications will comply in all respects with the 2004 revised Code of Practice.

2 Specification Aims

The aims of these AS GCE and Advanced GCE specifications are to encourage candidates:

- to develop an understanding of the main principles of solving problem using computers;
- to develop an understanding of the range of applications of computers and the effects of their use;
- to develop an understanding of the organisation of computer systems including software, data, hardware, communications and people;
- to acquire the skills necessary to apply this understanding to developing computer-based solutions to problems.

In addition, the Advanced GCE specification encourages candidates:

- to develop an understanding of the main principles of systems analysis and design, methods of problem formulation and planning of solutions using computers, and systematic methods of development, testing, implementation and documentation;
- to develop their capacity for critical thinking, see relationships between different aspects of the subject and to perceive their field of study in a broader perspective;
- to develop their project management skills and understanding of the need for team working.

2.1 Spiritual, Moral, Ethical, Social, Legal and Cultural Issues

These specifications encourage candidates to explore the spiritual, moral, ethical, social, legal and cultural aspects of the introduction of computer-based solutions to problems through a study of the effects on society.

Through candidates’ study of Module 2508 (see section 5.3.6), they have an opportunity to develop their understanding of spiritual, moral, ethical, social, legal and cultural issues. Section 5.3.6 considers issues such as changing leisure patterns and work practices, privacy and confidentiality of data held in systems, opportunities for access to information and environmental issues.

Module 2511 introduces a global dimension with the study of computing through electronic communications, for example, the Internet.

2.2 Environmental Education, the European Dimension and Health and Safety Issues

These areas are covered within Module 2508 (see section 5.3.6).

2.3 Avoidance of Bias

OCR has taken great care in the preparation of these specifications and assessment materials to avoid bias of any kind.

3 Assessment Objectives

AS GCE and Advanced GCE Computing have the same assessment objectives. In the Advanced GCE specification, the assessment objectives related to the skills of analysing, designing, development, testing, implementation, documentation and evaluating systems are given a higher weighting than in the AS GCE because of the increased emphasis on candidates developing their own computer-based solutions to real problems.

Knowledge, understanding and skills in computing are closely linked. Candidates should demonstrate the following assessment objectives in the context of the content and skills prescribed.

AO1 Knowledge and Understanding

Candidates should be able:

- to describe and explain the use and impact of computing in a range of applications and show an understanding of the characteristics of computer systems (hardware, software and communications) which allow effective solutions to be achieved;
- to describe and explain the need for, and the use of, various forms of data organisation and processing to support the information requirements of a particular application;
- to describe and explain the systematic development of effective solutions to problems, and the techniques appropriate for implementing such solutions;
- to comment critically on the spiritual, moral, ethical, social, legal, cultural and other consequences of the use of computers;
- communicate in writing:
  - to select and use a form and style of writing appropriate to purpose and complex subject matter;
  - to organise relevant information clearly and coherently, using specialised vocabulary where appropriate;
  - to ensure that text is legible, and spelling, punctuation and grammar are accurate, so that meaning is clear.

AO2 Skills

Candidates should be able:

- to analyse a problem and identify the parts which are appropriate for a computer-based solution;
- to select, justify and apply appropriate techniques and principles in developing data structures and algorithms for the solution of problems;
- to design and implement effective solutions using appropriate hardware and software.

The assessment objectives are weighted as follows:

	AS GCE	A2	Advanced GCE
AO1	58%	34%	46%
AO2	42%	66%	54%

3.1 Specification Grid

The relationship between assessment objectives and the units of assessment is shown in the specification grid below.

Unit	Level	Percentage of Advanced GCE		Total
		AO1	AO2	
2506	AS	12	3	15
2507	AS	5	15	20
2508	AS	12	3	15
2509	A2	8	7	15
2510	A2	2	18	20
2511	A2	7	8	15
Total		46	54	100

3.2 Quality of Written Communication

The Quality of Written Communication is assessed in respect of each of the units assessed through written papers, that is, in Units 2506, 2508, 2509 and 2511. The question paper rubric will provide information to candidates as to which questions include marks for the Quality of Written Communication. Assessment Objective 1 includes the assessment of Written Communication and candidates are required:

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

4 Scheme of Assessment

Candidates take three units for AS GCE, followed by a further three units at A2 if they are seeking an Advanced GCE award.

Units of Assessment

Unit	Level	Name	Mode of Assessment	Duration	Weighting	
					AS	Advanced GCE
2506	AS	Introductory Computer Systems, Communications and Software	Written Examination	1 hour 30 mins	30%	15%
2507	AS	Structured Practical Computing Tasks	Coursework	-	40%	20%
2508	AS	Computer Systems Development and Practical Applications	Written Examination	1 hour 30 mins	30%	15%
2509	A2	Systems Software Mechanisms, Machine Architecture, Database Theory and Programming Paradigms	Written Examination	1 hour 30 mins	-	15%
2510	A2	Computing Project	Coursework	-	-	20%
2511	A2	Integrated Information Systems	Written Examination	1 hour 30 mins	-	15%

Rules of Combination

Candidates must take the following combination of units.

AS GCE	Units 2506, 2507 and 2508
Advanced GCE	Units 2506, 2507, 2508, 2509, 2510 and 2511

Unit availability

There are two unit sessions each year, in January and June.

The availability of units is shown below.

Unit	Level	Unit Title	Jan 2005	June 2005
2506	AS	Introductory Computer Systems, Communications and Software	✓	✓
2507	AS	Structured Practical Computing Tasks	✓	✓
2508	AS	Computer Systems Development and Practical Applications	✓	✓
2509	A2	Systems Software Mechanisms, Machine Architecture, Database Theory and Programming Paradigms	✓	✓
2510	A2	Computing Project	✓	✓
2511	A2	Integrated Information Systems	✓	✓

The availability shown for 2005 will be the same in subsequent years.

Sequence of Units

The normal sequence in which the units could be taken is Units 2506, 2507 and 2508 in the first year of study leading to an AS GCE award, then Units 2509, 2510 and 2511 leading to an Advanced GCE award. However, the units may be taken in other sequences.

Alternatively, candidates may take all units at the end of their AS GCE or Advanced GCE course in a ‘linear’ manner, if desired.

Synoptic Assessment

Synoptic assessment tests candidates’ understanding of the connections between the different elements of the subject. It accounts for at least 20% of the total Advanced GCE marks and occurs particularly in the assessment of Unit 2511 (15%) and also in Unit 2510, in sections on analysis and evaluation (6%).

It is no longer a requirement to take synoptic units at the end of the course.

Certification

Candidates may enter for:

- AS GCE certification;
- AS GCE certification, bank the result, and complete the A2 assessment at a later date;
- Advanced GCE certification;

Candidates must enter all six AS and A2 units to qualify for the full Advanced GCE award.

Individual unit results, prior to certification of the qualification, have a shelf life limited only by that of the qualification.

Re-sits of Units

The restrictions on re-sitting units have been removed, enabling candidates to re-take units more than once. Upon making an entry for certification, the best attempt will be counted towards the final award. This change applies to all candidates, including those who have already been entered for any units or full qualifications.

Re-sits of AS GCE and Advanced GCE

Candidates may still enter for the full qualification an unlimited number of times.

4.1 Question Papers Units 2506, 2508, 2509 and 2511 are assessed by a written examination.

All question papers for these four units are of 1 hour 30 minutes duration. For all four units, there is a maximum of 90 marks, of which 4 marks are allocated to the assessment of the Quality of Written Communication.

4.1.1 AS

Unit 2506 - Introductory Computer Systems, Communications and Software (90 marks)

Assessment is by a written examination of 1 hour 30 minutes duration, containing 10-15 compulsory structured questions requiring short/paragraph length answers. Questions are **not** based in a context.

Unit 2508 - Computer Systems Development and Practical Applications (90 marks)

The purpose of this unit is to take the concepts developed in Module 2506 and relate them to the use of computer systems. In addition some new concepts are introduced; the intention being that they should arise through the study of computer applications.

Assessment is by a written examination of 1 hour 30 minutes duration with 6-8 compulsory structured questions based on short scenarios.

4.1.2 A2

Unit 2509 - Systems Software Mechanisms, Machine Architecture, Database Theory and Programming Paradigms (90 marks)

The assessment of this unit is by a written examination of 1 hour 30 minutes duration.

Candidates are also expected to have a general knowledge of information applications and to be able to apply such knowledge to specific examples. This unit includes reference to Modules 2506 and 2508, extending basic theory by considering topics such as Operating Systems, Databases and Programming Paradigms in greater depth. The examination paper contains 6-8 compulsory questions, some of which will relate to a context.

Unit 2511 - Integrated Information Systems (90 marks)

The assessment of this unit is by a written examination of 1 hour 30 minutes duration. The question paper contains 5-7 compulsory questions set in the context of a short scenario.

4.2 Coursework

There are two coursework units: Units 2507 and 2510.

4.2.1 AS

Unit 2507 - Structured Practical Computing Tasks (120 marks)

This unit examines knowledge and understanding as well as skills. The OCR-set tasks (between 3-5 tasks) are intended to allow candidates to demonstrate their competence in the skills of design, development, testing and documentation See section 5.2 for further information.

The tasks are marked by the teacher (mark schemes are sent to Centres after the date of final entries) and externally moderated by OCR.

OCR supplies general procedures, instructions and forms for moderation.

4.2.2 A2

Unit 2510 – Computing Project (120 marks)

This unit assesses candidates’ ability to develop a computer-based solution to a real-life problem requiring the skills of analysis, design, development, testing, implementation, documentation and evaluation. Candidates should formulate the task in negotiation with their teacher. If Centres are uncertain about the appropriateness of a problem, they should seek advice from OCR.

4.2.3 *Assessment and Moderation*

All coursework is marked by the teacher and internally standardised by the Centre. Marks are then submitted to OCR by the specified date, after which postal moderation takes place in accordance with OCR procedures. The purpose of moderation is to ensure that the standard for the award of marks in coursework is the same for each Centre, and that each teacher has applied the standards appropriately across the range of candidates within the Centre.

4.2.4 *Minimum Coursework Requirements*

If a candidate submits no work for a coursework unit, then the candidate should be indicated as being absent from that unit on the coursework mark sheets submitted to OCR. If a candidate completes any work for the coursework unit, then the work should be assessed according to the criteria and marking instructions, and the appropriate mark awarded, which may be 0 (zero).

4.2.5 *Authentication*

As with all coursework, the teacher must be able to verify that the work submitted for assessment is the candidate’s own work. Sufficient work must be carried out under direct supervision to allow the teacher to authenticate the coursework marks with confidence.

4.3 Special Arrangements

For candidates who are unable to complete the full assessment or whose performance may be adversely affected through no fault of their own, teachers should consult the *Inter-Board Regulations and Guidance Booklet for Special Arrangements and Special Consideration*. In such cases, advice should be sought from OCR as early as possible during the course. Applications for special consideration in coursework units should be accompanied by Coursework Assessment Forms giving the breakdown of marks for each skill.

4.4 Differentiation

In the question papers, differentiation is achieved by setting questions which are designed to assess candidates at their appropriate levels of ability and which are intended to allow all candidates to demonstrate what they know, understand and can do.

In coursework, differentiation is by task and by outcome in the Structured Practical Computing Tasks and by outcome in the Computing Project. Candidates should undertake coursework that enables them to display positive achievement.

4.5 Awarding of Grades

The AS has a weighting of 50% when used in an Advanced GCE award. An Advanced GCE award is based on the certification of the weighted AS (AS) (50%) and A2 (50%) marks.

Both AS GCE and Advanced GCE qualifications are awarded on the scale A to E and U (unclassified).

4.6 Grade Descriptions

The following grade descriptions indicate the level of attainment characteristic of the given grade at Advanced GCE. They should be interpreted in relation to the content outlined in the specification; they are not designed to define that content. They give a general indication of the learning outcomes and levels of attainment likely to be shown by a representative candidate performing within each stated grade. In practice, most candidates will show uneven profiles across the attainments listed, with strengths in some areas compensating in the award process for weaknesses or omissions elsewhere.

Grade A

Candidates demonstrate

- a good understanding of theoretical concepts;
- an appropriate and accurate use of technical language;
- a detailed knowledge of a range of applications;
- informed opinions on the effects of computing on society;
- the application of knowledge and understanding to unfamiliar problems;
- a good understanding of data types and structures, and how to use them ;
- an effective and appropriate use of a range of software;
- the ability to design and produce effective solutions to complex problems;
- a methodical, analytical and critical approach to problem solving;
- the ability to design, operate and justify appropriate testing strategies;
- clear communication of design decisions and solutions to problems;
- effective skills of evaluation.

Grade C

Candidates demonstrate

- an understanding of the main theoretical concepts;
- accurate use of technical terms;
- knowledge of a range of applications and their main effects on society;
- the application of knowledge and understanding to familiar problems;
- an understanding of data types and structures and their uses;
- the ability to use a range of software;
- the ability to design and produce solutions to substantial problems;
- a methodical and analytical approach to problem solving;
- the ability to design and operate appropriate testing strategies;
- clear communication of solutions to problems;
- skills of evaluation.

Grade E

Candidates demonstrate

- some relevant knowledge and limited understanding of theoretical concepts;
- use of basic technical terms;
- a basic knowledge of computing applications and their effects;
- recognition of hardware and software elements appropriate to a given situation;
- some knowledge of data types and structures;
- a basic use of analytical methods to solve straightforward familiar problems;
- some limited skill in justifying or considering alternatives;
- the ability to test solutions in a limited way;
- the ability to produce basic documentation;
- basic evaluative skills.

5 Specification Content

These specifications are set out in the form of teaching modules. Each teaching module is assessed by its associated unit of assessment.

Throughout this section the symbol 🔑 is used in the margin to highlight where Key Skills development opportunities are signposted. For more information on Keys Skills coverage please refer to Appendix A.

This section of the specifications gives the subject content for each module, as shown below.

Module	**Module Title**
2506	Introductory Computer Systems, Communications and Software
2507	Structured Practical Computing Tasks
2508	Computer Systems Development and Practical Applications
2509	Systems Software Mechanisms, Machine Architecture, Database Theory and Programming Paradigms
2510	Computing Project
2511	Integrated Information Systems

Each module is presented as a set of sub-sections, each with details of content and associated learning outcomes. An indication of recommended prior knowledge is given for each module, together with details of any links to other modules.

Module 2506: Introductory Computer Systems, Communications and Software is the foundation for all subsequent modules. It provides candidates with an understanding of the core aspects of computer systems, which is developed and enhanced in subsequent modules.

Module 2507: Structured Practical Computing Tasks provides candidates with opportunities to apply skills, knowledge and understanding from Modules 2506 and 2508 in a practical way. It is envisaged that work on the structured tasks will start almost in parallel to work on Module 2508, thus as soon as the relevant material in Module 2508 has been covered, it is advisable for candidates to begin the appropriate exercise for Module 2507.

The structured practical computing tasks examine separately the different phases in the development of a computer-based system; thus, they prepare candidates for the Computing Project, in the second half of the Advanced GCE, which requires the integration of these elements in response to an identified need. One or more tasks require the use of a computer system in their solution: in such cases, candidates may choose a solution based on the use of a programming language or an appropriate applications package.

Module 2508: Computer Systems Development and Practical Applications provides candidates with further ability to extend skills, knowledge and understanding of computing concepts, gained in Module 2506, to a range of applications in which computer systems are used.

Module 2509: Systems Software Mechanisms, Machine Architecture, Database Theory and Programming Paradigms Theory extends knowledge and understanding of concepts related to computer systems. Some concepts previously encountered in Modules 2506 and 2508 are developed in greater depth, while some new concepts are introduced. In addition, applications based on information provision are covered in this module.

Module 2510: Computing Project requires candidates to identify a well-defined user-driven problem, involving a third-party user, and to generate a solution. As for Module 2507, this is done using software tools chosen by the candidate and may include a programming language, an appropriate applications package or other software. It is envisaged that work on the Project will begin in parallel with work on Module 2509.

Module 2511: Integrated Information Systems applies and extends knowledge of systems and data and the development, implementation and management and applications of systems. The module covers the applications of real-time computing and simulation.

5.1 Module 2506: Introductory Computer Systems, Communications and Software

🔑 **C3.1a, C3.1b, C3.3; IT3.1**

This module provides candidates with an understanding of the following core aspects of computer systems:

- Components of a Computer System and Modes of Use;
- System Software;
- Programming Tools and Techniques;
- Data: Its Representation, Structure and Management;
- Hardware;
- Data Transmission and Networking.

This module is self-contained and assumes no previous knowledge, although the GCSE Information Technology syllabus covers some of this introductory material. For this reason the module may seem comparatively large.

The content of this module provides a foundation of computing skills, knowledge and understanding for all other modules.

5.1.1 Components of a Computer System and Modes of Use

Content

- Types of hardware
- Types of software
- Modes of use: batch, real-time, on-line, off-line

Learning outcomes

Candidates should be able to:

- define the terms: hardware, software, input device, storage device and output device;
- describe the purpose of input devices, storage devices and output devices;
- define the different types of software: operating system, user interface, translator, utilities, programming languages and generic/common applications software;
- describe the different roles and functions of systems software and applications packages;
- describe the characteristics of different modes of computer system use: batch, real-time, on-line and off-line.

5.1.2 System Software

Content

- Operating systems
- User interfaces
- Utilities

Only a basic introduction to operating systems and user interfaces is required since modules 2508 and 2509 cover these topics in more depth.

Learning outcomes

Candidates should be able to:

- describe the purpose of operating systems;
- describe the characteristics of different types of operating systems and their uses : batch, real-time, single-user, multi-user, multi-tasking and distributed systems;
- identify and describe the purpose of different types of user interface: forms, menus, GUI, natural language and command line, suggesting the characteristics of user interfaces which make them appropriate for use by different types of user;
- identify and describe the purpose of a range of utilities, for example, compression software.

5.1.3 Programming Tools and Techniques

Content

- Problem-solving techniques
- Features of procedural programming languages
- Basic translation process
- Program testing
- Program maintenance

Learning outcomes

Candidates should be able to:

- solve problems in a structured way, using logic and reason;
- describe techniques for writing software, including the splitting up of a problem into small sections and the use of appropriate techniques showing step-wise refinement/top-down design;
- identify and use programming constructs (sequence, selection, repetition);
- identify and use recursion;
- define and use procedures and functions;
- explain that translators are needed to convert source code to object code (detailed knowledge of the types of translator and the translation process is **not** required);
- describe and give examples of types of programming error (syntax, logic and arithmetic);
- design a test plan using different testing strategies such as white box testing, black box testing, alpha and beta testing ;
- describe and use appropriately the tools, techniques and methods available for identifying programming errors (translator diagnostics, debugging tools, desk checking, bottom up programming, test strategies);
- demonstrate an understanding of the need for good program design techniques to facilitate the ongoing maintenance of programs, for example, the use of comments, meaningful data names, indentation, and modularity.

5.1.4 Data: Its Representation, Structure and Management

Content

- Number systems
- Data types
- Data structures
- Data management

Learning outcomes

Candidates should be able to:

- express numbers in binary, binary coded decimal (BCD), octal and hexadecimal;
- describe and use two's complement and sign and magnitude to represent negative integers;
- perform integer binary arithmetic: addition and subtraction;
- explain the use of codes to represent a character set (e.g. ASCII and EBCDIC);
- define and use different data types: numeric (integer, floating point), Boolean and character;
- define and use arrays (single and multi-dimensional) for solving simple problems, including initialising arrays, reading data into arrays and performing a simple serial search on an array;
- define and use linked lists (single pointer) for solving simple problems, including initialising linked lists and performing a simple serial search on a linked list (detailed algorithms are not expected);
- describe the features of the data structures stacks and queues, including details of the way they operate as LIFO and FIFO structures;
- explain how data is stored in files in the form of fixed length records comprising items in fields;
- design a record format;
- estimate the size of a file;
- define and explain the difference between serial, sequential, indexed sequential and random access to data, using examples and stating their comparative advantages and disadvantages;
- describe how serial, sequential and random organisation of files may be implemented using indexes, and hashing algorithms as appropriate (detailed algorithms and procedures will not be required);
- select appropriate data types/data structures for a given problem and explain the advantages and disadvantages of alternative choices;
- explain the procedures involved in backing up data and archiving, including the difference between data that is backed up and data that is archived.

5.1.5 Hardware

Content

- Processor components
- Primary and secondary storage
- Peripheral devices

Learning outcomes

Candidates should be able to:

- describe the function and purpose of the control unit, memory unit and arithmetic logic unit (ALU) as individual parts of a processor;
- explain the difference between types of primary memory and their uses (RAM, ROM);
- describe the basic features, advantages, disadvantages and use of secondary storage media, both magnetic and optical;
- describe the transfer of data between secondary storage and primary memory, including the uses of buffers and interrupts;
- describe a range of common peripheral devices in terms of their features, advantages, disadvantages and uses: e.g. bar code readers, MICR, OCR, scanners, printers, plotters and microphones (this list is indicative and any device which could be connected to the computer for input, output and storage should be considered as included).

5.1.6 Data Transmission and Networking

Content

- Data transmission
- Circuit switching and packet switching
- Protocols
- Networking

Learning outcomes

Candidates should be able to:

- describe the characteristics of a local area network (LAN) and a wide area network (WAN);
- show an understanding of the hardware and software needed for a local area network (LAN) and for accessing a wide area network (WAN), for example, to the Internet;
- describe basic network topologies (bus, star and ring) explaining the benefits and drawbacks of each topology;
- describe the different types of data transmission: serial and parallel; and simplex, half duplex and duplex modes;
- explain the relationship between bit rates and information content;
- recognise that errors can occur in data transmission, and explain methods of detecting and correcting these errors (parity checks and check sums);

- explain the difference between packet switching and circuit switching;
- define the term protocol and name common examples;
- describe the need for communication between devices, and between computers, and explain the need for protocols to establish communication links (candidates will not be expected to have detailed knowledge of specific protocols);
- explain the principle of organising interfaces in a layered fashion to facilitate intercommunication;
- identify the advantages and disadvantages of networking.

5.2 Module 2507: Structured Practical Computing Tasks

C3.3; N3.2, N3.3; IT3.3	**WO3.2; PS3.1, PS3.2, PS3.3**

This module is designed to allow candidates to develop the following skills:

- Design;
- Software Development;
- Testing;
- Implementation.

The module covers basic knowledge and understanding, as well as skills. No one task tests all skills. One task will address at least two of these skills, and may involve the development of an algorithm.

- One task may involve the derivation and production of a testing strategy.
- Another may involve the design of a system perhaps to the production of pseudocode or other design description methods. Another may involve the understanding of pseudocode provided.
- One or more tasks involve the use of hardware for implementation.

It is expected that candidates will have studied all of Module 2506, requiring basic knowledge, and more particularly, programming skills, in order to carry out the tasks successfully. Additionally, some of the necessary sections of Module 2508 should have been studied, in particular the systems cycle. It is envisaged that work on the structured practical tasks will start almost in parallel to work on Module 2508, thus as soon as the relevant material in Module 2508 has been covered, it is advisable for candidates to begin the appropriate exercise in Module 2507.

In practising for the structured tasks, candidates may work in project teams and gain experience valuable for key skills, especially for *Working with Others*.

5.2.1 Design

Candidates should be able to specify and document a design, using appropriate algorithms/models. The design specification may include the method of solving a problem, for example:

- Hardware and software requirements;
- Input design;
- Output design;
- Data structures/model;
- Process model.

Learning Outcomes

Candidates should be able to:

- specify the required hardware for a given problem (related in particular to sections 5.1.1, 5.1.5);
- specify the required software for a given problem (related in particular to sections 5.1.2, 5.1.3, 5.1.6, 5.3.2);
- design and document data capture forms and/or screen layouts for a given problem (related in particular to sections 5.1.2, 5.3.3, 5.3.4);
- design and document report layouts, screen displays and/or other forms of output (for example, audio output) for a given problem, (related in particular to sections 5.1.5, 5.3.3);
- design and document the data structures necessary to model a given problem (related in particular to section 5.1.4);
- design and document an algorithm/pseudocode/top down diagram or other form of process model (related in particular to sections 5.1.3, 5.3.1).

5.2.2 Software Development

- Interpreting a design solution
- Developing a software solution

Learning Outcomes

Candidates should be able to:

- interpret a design solution specifying any variables and data structures using an appropriate software tool (related in particular to sections 5.1.3, 5.1.4, 5.3.2, 5.3.3);
- develop the rules/methods/algorithms of a design using a software package or programming language (related in particular to sections 5.1.3, 5.3.1, 5.3.2, 5.3.3);
- develop inputs/outputs using the features of a software package or a programming language (related in particular to sections 5.1.3, 5.1.2, 5.3.3, and 5.3.4).

5.2.3 Testing

- Test strategy
- Test data

Learning outcomes

Candidates should be able to:

- Identify, develop and document a test strategy for a given problem (related in particular to section 5.1.3);
- Select suitable test data for a given problem (related in particular to section 5.1.3).

5.2.4 Implementation

- Testing a software solution and planning for its implementation
- User instructions

Learning outcomes

Candidates should be able to:

- test a software solution, providing documented evidence that the solution works and plan for its implementation (related in particular to sections 5.1.3, 5.3.1);
- prepare basic user instructions for the software solution (related in particular to sections 5.1.3, 5.3.1).

5.3 Module 2508: Computer Systems Development and Practical Applications

🔑	**C3.1a, C3.1b, C3.2, IT3.1, IT3.3**	

This module relates the concepts developed in module 2506 to the practical use of computer systems.

The systems development life cycle is studied with reference to particular applications. Therefore, candidates are expected to look at a range of different types of application areas. Although candidates are not expected to have specific knowledge of every one, candidates should be able to make use of relevant examples for the purpose of illustration.

- Systems Development Life Cycle
- Choosing Applications Software for Application Areas
- Handling of Data in Information Systems
- Designing the User Interface

- Characteristics of Information Systems
- Implications of Computer Use

Recommended Prior Knowledge

Coverage of the content of Module 2506 is assumed in Module 2508.

5.3.1 *Systems Development Life Cycle*

Content

- Identification of problem
- Feasibility study
- Information collection
- Analysis of a problem, based upon information collected, including producing a requirements specification
- Design of system to fit requirements
- Development and testing of system
- Implementation of system

Learning outcomes

Candidates should, with reference to particular applications, be able to:

- explain the importance of defining a problem accurately;
- describe the function and purpose of a feasibility study;
- explain the importance of determining the information requirements of a system and describe different methods of fact finding, including questionnaires, observation, and structured interviews, highlighting the advantages and disadvantages of each method;
- describe what is involved when analysing the requirements of a system, explaining the nature of the requirements specification and its content: including current data structures, inputs, outputs and processing represented in diagrammatic form (data flow diagrams, system flowcharts or Jackson diagrams), identify inefficiencies/problems, user requirements and hardware and software requirements;
- describe what is involved when designing a system, including input design, diagrammatic depiction of the overall system processing (see above), data structure design and output design;
- explain the importance of evaluating the system, and how to identify the criteria used for evaluation;
- explain the content and importance of different types of documentation at different stages in the system life cycle, including the requirements specification, design specification, program specifications, technical and user manuals;
- explain the importance of system testing and implementation planning;
- explain the purpose of maintaining the system, and explain the need for system review and reassessment, understanding that software has a limited life span.

5.3.2 Choosing Applications Software for Application Areas

Content

- Custom written software versus off-the-shelf software package
- Application areas
- Applications software

Learning outcomes

Candidates should, within context, be able to:

- distinguish between custom written software and off-the-shelf software packages, and discuss the advantages and disadvantages of both in given situations;
- identify the features of common applications found in business, commercial and industrial applications: e.g. stock control, order processing, payroll, process control, point of sale systems, marketing, computer aided design (CAD), computer aided manufacture (CAM);
- identify suitable common generic applications software for particular application areas e.g. word processing, spreadsheets, desktop publishers (DTP), presentation software, drawing packages;
- identify application areas for which common applications software is not appropriate;
- describe the purpose and impact of different types of common/generic applications software for example, word processing, spreadsheets, desktop publishers (DTP), presentation software, drawing packages.

5.3.3 Handling of Data in Information Systems

Content

- Data capture, preparation and data entry
- Validation and verification of data
- Outputs from a system

Learning outcomes

Candidates should, within a context, be able to:

- describe manual and automatic methods of gathering and inputting data into a system, including form design, keyboard entry, voice recognition, barcodes, optical mark recognition, magnetic stripe cards, (OMR), optical character recognition (OCR), data logging, touch screens;
- describe image capture by use of a scanner, video capture card and digital camera/camcorder;
- distinguish between free text and structured data, transaction data and data prepared off line, describing their characteristics and uses;
- explain the techniques of validation and verification, and describe validation tests which can be carried out on data;

- describe possible output formats such as graphs, reports, interactive presentations, sound, video, images, animations stating the advantages and disadvantages of each format;

- discuss the need for a variety of output formats according to the target audience. Knowledge of timeliness, relevance etc. of intended output is required.

5.3.4 Designing the User Interface

Content

- Interface design

- Criteria for selecting appropriate hardware

- Interface style

Learning outcomes

Candidates should be able to:

- discuss the importance of good interface design;

- select appropriate peripheral hardware, including special purpose input devices, for a given application, and justify the choice;

- discuss human computer interface (HCI) design issues such as short-term and long-term memory, visual perception, use of colour, layout, and content;

- distinguish between different styles of interface, including forms, menus, command-line input, natural language, speech, direct manipulation, and their relevance to application design;

- identify the required characteristics of a user interface with respect to information, type of user, physical location and current technology;

- understand the potential problem of speed mismatch between user, peripheral and processor.

5.3.5 Characteristics of Information Systems

Content

- Passive versus interactive systems

- Characteristics and uses of management information systems

- Batch processing and rapid response applications

- Knowledge-based systems (expert systems)

Learning outcomes

Candidates should, within a context, be able to:

- give examples of, and describe the differing characteristics of, passive information systems and interactive information systems;
- describe the characteristics and uses of management information systems (MIS);
- identify a range of applications requiring batch processing and applications in which a rapid response is required;
- describe the characteristics of knowledge-based systems;
- identify and give examples of the different categories/types of knowledge-based systems such as diagnostic, advice giving, decision making;
- explain the difference between business MIS and knowledge-based systems.

5.3.6 Implications of Computer Use

Content

- Economic implications
- Social implications
- Legal implications
- Ethical implications
- Environmental implications

Learning outcomes

Candidates should be able to:

- discuss changing trends in computer use and their economic, social, legal and ethical effects on society;
- explain changes to society brought about by the introduction and use of computer systems, for example, in changing leisure patterns and work expectations;
- discuss the effects on privacy and confidentiality of data held in computer systems, and steps which can be taken to protect confidentiality;
- understand the need for, and the measures contained in, current legislation governing computer use, including Data Protection legislation and the European Directives;
- discuss the social and ethical implications of access to information whose value is controversial;
- discuss health and safety implications (for example, repetitive strain injury (RSI)) of increased computer use, including measures to ensure a healthy and safe working environment for employees;
- discuss the environmental issues concerned with the use of computers.

5.4 Module 2509: Systems Software Mechanisms, Machine Architecture, Database Theory and Programming Paradigms

C3.1b, C3.3.

This module is concerned with how computer architecture and systems software (such as operating systems and translators) function together. Additionally, a study of a range of programming paradigms and their target applications are included, together with database theory. The depth of knowledge candidates will acquire is extensive and extends the AS level initial studies in these areas, covered primarily by Module 2506. The content includes:

- The Function of Operating Systems
- The Function and Purpose of Translators
- Computer Architectures and the Fetch-Execute Cycle
- Data Representation, Data Structures and Data Manipulation
- Programming Paradigms
- Databases

Recommended Prior Knowledge

Candidates should have studied Module 2506.

5.4.1 The Function of Operating Systems

Content

- Features of operating systems
- Scheduling
- Interrupt handling
- Job queues and priorities
- Memory management
- Spooling
- Modern personal computer operating systems

Learning outcomes

Candidates should be able to:

- describe the main features of operating systems: for example, memory management, scheduling algorithms;
- explain how interrupts are used to obtain processor time and how processing of interrupted jobs may later be resumed, (typical sources of interrupts should be identified and any algorithms and data structures should be described);
- define and explain the purpose of scheduling, job queues, priorities and how they are used to manage job throughput;
- explain how memory is managed in a typical modern computer system, (virtual memory, paging and segmentation should be described along with some of the problems which could occur such as disk threshing);
- describe spooling, explaining why it is used;
- describe the main components of a typical desktop PC operating system, including the file allocation table and how it is used, and the purpose of the boot file;
- describe the main components of a network operating system including transparency, directory services, security and network printing.

5.4.2 The Function and Purpose of Translators

Content

- Types of translators
- Lexical analysis
- Syntax analysis
- Code generation and optimisation
- Linkers and loaders

Learning Outcomes

- Candidates should be able to:
- describe the difference between interpretation and compilation;
- describe what happens during lexical analysis;
- describe what happens during syntax analysis, explaining how errors are handled;
- explain the code generation phase and understand the need for optimisation;
- explain the purpose of linkers and loaders

5.4.3 Computer Architectures and the Fetch-Execute Cycle

Content

- Von Neumann architecture
- Registers: purpose and use
- Fetch-execute cycle
- Other machine architectures

Learning outcomes:

Candidates should be able to:

- describe basic Von Neumann architecture, identifying the need for, and the uses of, special registers in the functioning of a processor;
- describe, in simple terms, the fetch/decode/execute/reset cycle, and the effects of the stages of the cycle on specific registers;
- discuss other machine architectures including parallel processor systems, their uses, advantages and disadvantages.

5.4.4 Data Representation, Data Structures and Data Manipulation

Content

- Floating point binary
- Normalisation of floating point binary numbers
- Implementation of data structures, including lists, stacks, queues and trees
- Searching and sorting

Learning Outcomes

Candidates should be able to:

- demonstrate an understanding of floating point representation of a real binary number;
- normalise a real binary number;
- discuss the trade-off between accuracy and range when representing numbers;
- explain the difference between static and dynamic implementation of data structures, highlighting the advantages and disadvantages of each;
- describe algorithms for the insertion, deletion and amendment of data items stored in list, stack, queue and tree structures;
- explain the difference between binary searching and serial searching, highlighting the advantages and disadvantages of each;
- explain the concept of algorithms for implementing a binary search and serial search;
- explain the difference between the sort/merge methods insertion, quick sort and merging, highlighting the characteristics, advantages and disadvantages of each;
- describe algorithms for implementing insertion, quick sort and merging methods;

5.4.5 Programming Paradigms

Content

- Types of languages and typical applications
- Features of different types of language
- Methods for defining syntax

Learning outcomes

Candidates should be able to:

- identify a variety of programming paradigms (low level, object-orientated, declarative, procedural and functional);
- explain with examples, the terms object-oriented, declarative, procedural, and functional as applied to high level languages, showing an understanding of typical uses;
- explain how functions, procedures and their related variables may be used to develop a program in a structured way, using stepwise refinement, Jackson structured programming (JSP) or other similar top-down techniques;
- describe the use of parameters, local and global variables as standard programming techniques;
- explain how a stack is used to handle procedure calling and parameter passing;
- discuss the concepts, and using examples, show an understanding, of data encapsulation, classes and derived classes, and inheritance when referring to object-oriented languages;
- discuss the concepts, and using examples, show an understanding, of backtracking, instantiation, predicate logic and satisfying goals when referring to declarative languages;
- discuss the concepts, and using examples, show an understanding, of list processing, head and tail recursion when referring to functional languages;
- explain the concepts, and using examples, demonstrate an understanding of the use of the accumulator, registers, program counter. Direct, indirect and indexed addressing of memory when referring to low level languages;
- using examples, describe the nature and purpose of 3rd and 4th generation languages;
- explain the need for, and be able to apply, BNF (Backus-Naur form) and syntax diagrams.

Note: Candidates will **not** be expected to use any particular form to present algorithms, but should be able to write procedural algorithms in some form.

Candidates will **not** be expected to write or interpret the meaning of simple segments of low level language code.

A detailed knowledge of the syntax of programming languages is **not** required.

5.4.6 Databases

Content

- Database design
- Normalisation and data modelling
- Methods and tools for analysing and implementing database design
- Control of access to relational database elements
- File access methods

Learning outcomes

Candidates should, in general and within a context of a scenario, be able to:

- describe flat files, network, hierarchical and relational databases;
- analyse and size a database application;
- design a simple relational database to the third normal form (3NF), defining tables and views of data
- draw entity-relationship (E-R) diagrams to represent diagrammatically the data model;
- design forms for input, deletion, modification and querying of a database;
- explain the advantages that using a relational database gives over flat files;
- utilise formalised techniques for designing databases: entity-relationships diagrams/entity life history;
- define, and explain the purpose of, primary, secondary and foreign keys;
- explain the importance of varying the access allowed to database elements at different times and for different categories of user;
- describe the structure of a data base management system (DMBS), including the function and purpose of the data dictionary, data description language (DDL) and data manipulation language (DML)

5.5 Module 2510: Computing Project

🔑

C3.2, C3.3; N3.1, N3.2, N3.3; IT3.1, IT3.2, IT3.3

WO3.1, WO3.2, WO3.3; LP3.1, LP3.2, LP3.3; PS3.1, PS3.2, PS3.3

This module allows candidates to develop their knowledge and understanding of computer systems and the skills described in Assessment Objective 2 (see Section 3). The project is a substantial piece of work requiring analysis and design over an extended period of time, which is organised, evaluated and presented in a report.

Candidates choose, in conjunction with their teacher, a well-defined user-driven problem of an appropriate size which enables them to demonstrate their skills in Analysis, Design, Software Development, Testing, Implementation, Documentation and Evaluation, and their interrelation, giving a completed overall task solving the problem.

Recommended Prior Knowledge

Candidates are likely to have completed Modules 2506, 2507 and 2508 before starting the computing project. It is unlikely that, without this or equivalent background, they will possess the necessary integrative skills, knowledge and understanding to attempt a computing project at Advanced GCE. Some sections of Module 2509 (for example, sections 5.4.4, 5.4.5 and 5.4.6) must also be studied before tackling the project but, on the whole, it is envisaged that the project will be conducted in parallel with the study of Modules 2509 and 2511.

Projects should be chosen to demonstrate the integrative aspects of the work and should avoid needless repetition of the demonstration of a given skill. Each candidate must submit a report on their piece of work, supported by evidence of software development and testing.

The teacher marks the projects using a mark scheme provided by OCR, after which moderation takes place according to OCR procedures.

Computing projects are assessed as follows:

• Definition, Investigation and Analysis	**[25 marks]**
• Design	**[21 marks]**
• Software Development , Testing and Implementation	**[35 marks]**
• Documentation	**[24 marks]**
• Evaluation	**[15 marks]**

Further details are given in Appendix B.

5.5.1 Definition, Investigation and Analysis 25 marks

Explanation of the problem to be solved, the user’s requirements and how they were obtained. There should be a clear statement of requirements.

- Define a problem
- Investigate the problem
- Record findings
- Analyse findings
- Identify problems/inefficiencies with current system
- Specify requirements: user, hardware, software

Learning outcomes

Candidates should be able to:

- define the nature of the problem to be solved (related to section 5.3.1);
- identify methods by which to investigate the problem: including questionnaires, observation and structured interviews (related to section 5.3.1);
- record information/data and gather sample documents currently used (related to sections 5.3.1, 5.3.3);
- identify the current processes and current data structures (related to sections 5.3.1, 5.1.3, 5.1.4, 5.3.3);
- analyse the data and processes: candidates will be expected to use appropriate techniques such as structure diagrams/data flow diagrams/system flowcharts to illustrate their analysis (related to sections 5.3.1, 5.4.4, 5.4.5);
- specify inefficiencies and problems apparent from discussions with the user and the analysis work carried out (related to section 5.3.1);
- derive the user and information requirements of a system (related in particular to section 5.3.1);
- specify the required hardware and reasons for choice (related in particular to sections 5.1.1, 5.1.5);
- specify the required software and reasons for choice (related in particular to sections 5.1.2, 5.1.3, 5.1.6, 5.3.2);
- develop and document a clear requirement specification (related to section 5.3.1).

5.5.2 Design 21 marks

Detailed system design including data structures, input-output format and processes involved. There should be a clear design specification.

- Input design
- Output design
- Data structures/model
- Process model

Learning Outcomes

Candidates should be able to:

- design and document data capture forms and/or screen layouts, drawing up detailed mock ups of the proposed interface (related in particular to sections 5.1.2, 5.3.3, 5.3.4);
- design and document report layouts, screen displays and/or other forms of output (for example, audio output), drawing up detailed mock ups of the proposed interface (related in particular to sections 5.1.5, 5.3.3);
- design and document (using appropriate techniques for example, normalisation/E-R models) the data structures necessary to solve the inefficiencies/problems indicated in the requirements specification (related in particular to sections 5.1.4, 5.4.5);
- design and document an algorithm/pseudocode/top-down diagram or other form of process model (related in particular to sections 5.1.3, 5.3.1, 5.4.4, and 5.4.5).

5.5.3 Software Development, Testing and Implementation 35 marks

A software solution and comprehensive test plan is developed from the design, which should show that the system works with valid, invalid and extreme data (or, if it does not, under which circumstances it fails.) The test plan should be clearly cross-referenced to evidence that the system has been tested during development and implementation. User testing should be in evidence.

- Software development
- Test strategy/plan
- Test data
- Testing a software solution and planning for its implementation
- User testing

Learning outcomes

Candidates should be able to:

- develop the rules/methods/algorithms of a design using a software package and/or programming language (related to sections 5.1.3, 5.3.1, 5.3.2, 5.3.3, 5.4.4, 5.4.5);
- develop the data structures of the design using the appropriate features of a software package and/or programming language (related to sections 5.1.4, 5.3.1, 5.3.2, 5.4.4, 5.4.6);
- develop inputs/outputs using the features of a software package and/or programming language (related to sections 5.1.3, 5.1.2, 5.3.3, and 5.3.4).
- identify, develop and document a test strategy for the design (related in particular to section 5.1.3);
- select suitable test data for the design (related in particular to section 5.1.3).
- test the software solution, illustrating how the software solution evolves (related to sections 5.1.3, 5.3.1);
- produce detailed output from the testing, cross referencing as appropriate the test plan (related to sections 5.1.3, 5.3.1);
- test the software solution with the user, providing documented evidence that the solution works and devise a strategy for its implementation (related to sections 5.1.3, 5.3.1).

5.5.4 Documentation 24 marks

- Technical documentation
- User guides

Technical Manual:

This should include an explanation of the structure of the design and the solution. All the necessary information about the system that would allow someone else to maintain and develop it should be included, for example, back up procedures/cycles, code/modules written and what they do, how they were written, data structures used and how they may be modified etc.

User Manual:

This should include step by step instructions for operating all aspects of the system, including a means of dealing with any errors that may occur. As well as a guide, User Documentation should include appropriate “Help” and messages within the software solution, or be present in the form of a hypertext document.

Learning outcomes

Candidates should be able to:

- develop a detailed technical manual (related to sections 5.1.3, 5.3.1);
- develop a detailed user manual (related to sections 5.1.3, 5.3.1).

5.5.5 Evaluation 15 marks

Discussion of the degree of success in meeting the original objectives as specified in the requirements specification, ease of the use of the package, acceptability to the users (including where possible a letter of acceptance from the user and reference to user testing results) and desirable extensions.

- Evaluate results against the requirement specification
- Evaluate user testing
- Identify the good and bad points of the final system, including any limitations and necessary extensions to the system

Learning outcomes

Candidates should be able to:

- evaluate the final system against the criteria described in the requirements specification (relates to section 5.3.1);
- evaluate the users’ responses to testing the system (relates to sections 5.3.1, 5.1.3);
- identify the good and bad points of the final system highlighting any limitations and necessary extensions to the system, indicating how the extensions could be carried out.

5.5.6 The Written Report

Candidates should submit a concisely written and well-laid out report, which should be word-processed. The selection of the problem for which a computerised system is to be designed and implemented is extremely important. It should be chosen by the candidate in consultation with the teacher, and should always involve a user, ideally a ‘third party’ user.

It is important to stress that the candidate should endeavour to produce a system which is non-trivial and which will solve a given problem sensibly within the constraints of resources available to the candidate. For example, an airline seat reservation system cannot normally sensibly be implemented on a stand-alone PC.

Since the computing project seeks to assess the systems analysis section of the specification in a practical manner, candidates should **not** produce a system from their own limited knowledge of the requirements of the system. The ‘third-party’ user has to be someone who is willing to be involved in the project

- in the analysis of the problem, where the user’s requirements are obtained. This may take the form of a recorded interview with the candidate.
- at the software development, testing and implementation stages, where the user is involved in ‘prototyping’.
- at the evaluation stage, where the user is involved in checking that the system is completed as specified and, leading on from this, is then willing to write a letter of acceptance of the system, including any criticisms of it.

In this way, candidates can be encouraged to look beyond school life into the businesses and companies in the community of the surrounding area. The emphasis is on analysing an existing system, and producing a computer-based solution to fit the needs of the user.

More detailed marking criteria for these stages are given in Appendix B.

5.6 Module 2511: Integrated Information Systems

🔑 **C3.1a, C3.1b, C3.2, C3.3; IT3.1, IT3.2**

This module is concerned with building on all the other modules studied in this specification. It introduces new areas of computing such as applications of real-time computing and simulation.

Module 2511 serves to integrate the concepts of the whole specification, and it includes the following issues, which are considered to be pervasive:

- The importance and use of standards, both formal and *de facto;*
- The importance and use of top-down design;
- The importance and process of prototyping;
- The need for training and retraining.

The assessment is synoptic, and all questions in Unit 2511 are asked in the context of a scenario, which may assume any aspect of knowledge and skills from other parts of the specification.

5.6.1 Use of Systems and Data

Content

- The commercial value of data
- The importance of standards
- Communications and electronic commerce
- Training
- Effects of introducing systems

Learning outcomes

Candidates should, within the context of a scenario, be able to:

- identify data which has commercial value, explaining why such data has this value, and discuss contemporary trends in the compilation and use of valuable databases;
- explain the advantages of standardisation and describe some areas of standardisation such as file formats, ISDN, OSI model and its use together with communications protocols;
- describe ways in which computers aid communication, including voice mail, e-mail, digital telephone system facilities, e-commerce over the internet, tele/videoconferencing and electronic data interchange;
- select and justify an appropriate network system for a particular application, including the bandwidth required to transmit different forms of data such as text, sound, real-time sampled data and video, hardware such as file servers, hubs, repeaters and switches;
- identify situations in which the transmission of data, for example over the Internet, has created or could create new opportunities for businesses and individuals, in particular explaining how e-commerce works;
- identify and describe training and re-training requirements for a given scenario;
- describe the substantial short-term and long-term changes, in patterns of work and in quality of output, which occur as a result of introducing computing systems.

5.6.2 Systems Development, Implementation, Management and Applications

Content

- Methodologies and software tools for system development
- Application types and technical requirements
- Choice of implementation approaches: direct, parallel, phased
- Project management and software tools
- Systems management and monitoring

Learning outcomes

Candidates should, within the context of a scenario, be able to:

- identify commonly used techniques and/or methodologies for developing computers systems such as entity models/SSADM;
- describe how use of methodologies/techniques and software tools for developing computer systems aid the systems analyst/designer and programmer in terms of the documentation, step-by-step logical progression through tasks and cross-checking mechanisms;
- discuss the technical requirements of a system necessary to implement a range of different computer applications, including hardware, operating systems, communications, interface software, and other utility software;
- explain the need to provide appropriate response times for different applications, in the context of a scenario, and its implications for hardware, software and data structures;
- select, plan and justify appropriate implementation approaches for a range of different applications such as: parallel, phased, pilot, direct;
- describe the process of project management and give examples of different aids a project manager may use to plan and monitor a project such as Gantt charts, critical path analysis, identifying the benefits and drawbacks of each;
- discuss, giving examples, the requirement for effective project management for the implementation of different computing applications, including the benefit of using project management software;
- discuss the implications of managing, monitoring and maintenance of systems, including the need for up-to-date documentation, software audit, quality control and management and hardware updates.

5.6.3 Simulation and Real-time Processing

Content

- Applications of real-time computing
- The feedback loop; input and output; sensors and actuators
- The use of robots
- Uses of simulation
- Variation of parameters and conditions; time steps
- Processing requirements
- Advantages and limitations of simulations

Learning outcomes

Candidates should, in general and within the context of a scenario, be able to:

- describe a range of real-time applications;
- explain the use of sensors and actuators for visible, tactile, audible and other physical signals;
- demonstrate an understanding of the use of robots in a variety of situations such as the manufacturing process or hazardous environments;
- explain the reasons for simulation, such as to change time-scales and/or save costs and/or avoid danger;
- describe the uses of simulation to assist in design, to make predictions, to test hypotheses;
- describe a simulation and its variables, the facility to vary conditions and observe the sensitivity of results to such variations;
- explain the large processing requirements of some such systems and hence recognise the need for parallel architectures;
- discuss the advantages of simulation in testing the feasibility of a design;
- discuss the limitations of simulation, especially where the situation is subject to random events.

5.6.4 Common Network Environments, Connectivity and Security Issues

Content

- Data transmission
- Network components
- Use of networks to support hyperlinking systems such as the world wide web (WWW)
- Common network environments
- Issues of confidentiality
- Encryption and authentication techniques

Learning outcomes

Candidates should, in general and within the context of a scenario, be able to:

- describe methods (analogue and digital) used to organise LANs and WANs, and typical rates of data transmission associated with different topologies and methods;
- demonstrate awareness of different media for transmitting data and their carrying capabilities;
- explain the different purposes of network components, including switches, routers, bridges and modems;
- discuss the demands of a nomadic network environment such as mobile phones;
- discuss common network environments, such as intranets, the Internet and open networks, their facilities, structure and ability to exchange information using appropriate software and techniques;
- describe the purpose of hypertext linking, identifying the means by which it can be achieved such as hotwords/links, buttons and hypertext mark up language (HTML);
- describe the basic features of mark up languages;
- describe the facilities provided by electronic mail systems (including voice mail) such as composing, responding, filing, copying, attaching, sending on and multiple recipients, and explain situations in which they are appropriate;
- discuss the problems associated with matching human and device speeds;
- describe how a network environment affects the user interface provided;
- discuss the problem of maintaining confidentiality of data on an open network and how to address this problem;
- explain the need for encryption, authorisation and authentication techniques (candidates will not be expected to know any specific method in detail);
- explain that distribution of a network can have implications for both data and responsibility;
- describe the uses of distributed databases, and understand the advantages and limitations of such distribution.

6 Further Information and Training for Teachers

To support teachers using these specifications, OCR will make the following materials and services available:

- a full programme of in-service training (INSET) meetings;
- specimen question papers and mark schemes;
- past question papers and mark schemes after each examination session;
- coursework guidance materials;
- written advice on coursework proposals;
- individual feedback to each Centre on the moderation of coursework;
- a Report on the Examination, compiled by senior examining personnel, after each examination session.

If you would like further information about these specifications, please contact OCR.

7 Reading List

This reading list is a compilation of books that have been recommended by teachers, as being useful for A Level Computing.

General

The first list of books cover the majority of the specification.

French, C.S.	*Understanding Computer Science for Advanced Level* (Third Edition). Stanley Thomas (Publishers) Ltd.
Heathcote, P.M.	*Active Learning Series: Computing etc*. DPP
Blackford & Leadbetter	*OCR AS & A Level Computing* Band L Publishing Co.
Cooper, D	*Computer Studies for A Level* ISBN, 0954351436
Bradley, R.	*Understanding Computer Science for Advanced Level* Third Edition
Bradley, R.	*New Understanding Computer Science for Advanced Level* (Third Edition). 1999. Stanley Thomas (publishers) Ltd. ISBN 0748740465
Heathcote, P.M.	*A Level Computing* 2004. Letts Educational Ltd. ISBN 1904467520

The next set of books are grouped within the units that they are mainly targeting.

Unit 1 Computing Fundamentals

Bailey, B.	*Data Structures* Blackie
Barker	*System Software* Blackie
Berk, Tony	*Essential Information Technology*
Jackson	*Approaches to Programming*
Fanning	*Programming Exercises Richard Ball Publishing,* Brassey St. Birkenhead
Mett, Percy	*Introduction to Computing* 1990. Macmillan Press. ISBN 0333393368

Unit 2 Structured Tasks

Heathcote, P.M. *Tackling Computer Projects in Access with Visual Basic*
(4nd Edition). 2004. Letts Educational. ISBN 1904467539

Heathcote, P.M. *Successful I.T Projects in Word*
1998. Payne-Gallway Publishers Ltd. ISBN 0953249042

Heathcote, P.M. *Successful I.T Projects in Excel*
1999. Payne-Gallway Publishers Ltd. ISBN 0953249050

Schneider, D. *Introduction to Programming with Visual Basic 6.0*
1999. Prentice Hall. ISBN 0139364285

Deitel, H. & Deitel, P. *Java: How to Program*
1997. Prentice Hall. ISBN 0138993947

Friedman, F.& Koffman, E. *Problem Solving, Abstraction, and Design Using C++*
(2nd Edition). 1997. Addison Wesley Longman Higher Education.
ISBN 0201883376

Horton, I *Beginning Visual C++5*
1998. Wrox Press Inc. ISBN 1861000081

Gurewich, N. & Gurewich, O. *Visual Basic 6 in 21 Days*

Mott, J. & Rendell, I *Database Projects in Access for Advanced Level*
Hodder & Stoughton. ISBN 034081201X

Mott, J. & Rendell, I *Database Projects in Excel for Advanced Level*
Hodder & Stoughton. ISBN 0340812028

Unit 3 Applied Computer Systems

Books in Unit 1 will be useful. The following give additional information.

Clifton, H.D. & Sutcliffe, A. *Business Information Systems (New Edition)*
1994. Prentice Hall Europe (Academic). ISBN 0131079700

Unit 4 Further Computing Theory

Books in Unit 1 will be useful. The following give additional information.

Sebesta,R.W. *Concepts of Programming Languages*
1998. Addison Wesley Longman Higher Education. ISBN 0201385961

Ben-Ari, M. *Understanding of Programming Languages*
1995. John Wiley and Sons Ltd. ISBN 0471958468

Date C.J. *Introduction to Database Systems*
1994. Addison Wesley World Student Series. ISBN 0201824582

Eaglestone, B.M. *Relational Databases*
1991. Stanley Thomas (Publishers) Ltd. ISBN 0748711767

Howe, D.R. *Data Analysis for Database Design*
1989. Arnold (Hodder & Stoughton Publishers). ISBN 071313688X

Sloane *Computer Communications (Principles and Business Applications)*
(2nd Edition). 1998.

Langfield, S *Learning to Program in Pascal and Delphi*
2003.Payne-Galloway Publishers. ISBN 1904467296

Unit 5 Project

As for Unit 2

Unit 6 Integrated Information Systems

Clifton, H.D. &

Sutcliffe, A. *Business Information Systems (New Edition)*
1994. Prentice Hall (Academic). ISBN 0131079700

Sloane *Computer Communications (Principles and Business Applications)*
(2nd Edition). 1998.

Appendix A

Key Skills

These specifications provide opportunities for the candidates to develop and generate evidence for assessing the Key Skills of *Communication, Application of Number, Information Technology, Working with Others, Improving Own Learning and Performance* and *Problem Solving* as required by QCA's subject criteria for Computing.

Through classwork, coursework and preparation for external assessment, candidates may produce evidence for these Key Skills at Level 3. However, the extent to which this evidence fulfils the requirements of the QCA Key Skills specifications at this level will be dependent on the style of teaching and learning adopted for each module. In some cases, the work produced may meet the evidence requirements of the Key Skills specifications at a higher or lower level.

Throughout section 5 the symbol 🔑 is used in the margin to highlight where Key Skills development opportunities are signposted. The following abbreviations are used to represent the above Key Skills:

C = Communication

N = Application of Number

IT = Information Technology

WO = Working with Others

LP = Improving Own Learning and Performance

PS = Problem Solving

These abbreviations are taken from the QCA Key Skills specifications for use in programmes starting from September 2000. References in section 5 and Appendix A, for example **IT3.1**, show the Key Skill (IT), the level (3) and subsection (1).

Centres are encouraged to consider the OCR Key Skills scheme to provide certification of Key Skills for their candidates.

Detailed opportunities for generating Key Skills evidence through this specification are posted on the OCR website, www.ocr.org.uk

Key Skills Coverage

The following matrix indicates those Key Skills for which opportunities for at least some coverage of the relevant Key Skills exist.

Module	Communication	Application of Number	IT	Working with Others	Learning Performance	Problem Solving
	Level 3	Level 3	Level 3	Level 3	Level 3	Level 3
2506	✓		✓		✓	
2507	✓	✓		✓	✓	✓
2508	✓		✓		✓	
2509	✓				✓	
2510	✓	✓	✓	✓	✓	✓
2511	✓		✓		✓	

Appendix B

Guidance on Setting and Marking the Computing Project (Unit 2510)

Guidance on Setting Computing Projects

A Project should:

- allow candidates to demonstrate their knowledge and understanding of computer systems and the skills in the assessment objectives;
- encourage the sensible use of computers to produce a system, which is non-trivial and solves a problem sensibly, within the constraints of resources available to candidates. Prototypes of larger systems are not of themselves likely to cover the required skills;

It may be difficult to quantify the scale of a problem fully. Indeed, before the analysis and design stages have been completed it is not possible to know the depth of the solution. Candidates should note that the emphasis should be on the choice of a real life problem. Centres are reminded that if they are in any doubt about the suitability of a problem on the grounds of degree of rigour, type of problem or any other criteria, they should contact the board at the earliest opportunity for advice;

- show the successful completion of a whole task from definition involving a third party to acceptance and evaluation by that user. Projects, which involve much repetitive design, analysis or especially implementation, leading to unwieldy reports, are to be discouraged;
- involve all elements of the skills of definition, analysis, design, development, testing, implementation, documentation and evaluation. Projects need not be ‘stand alone’: the enhancement or modification of an existing system are acceptable, provided that all these elements are covered, this may in fact lead to work that is more likely to reflect a real world situation;
- involve a third party user, who will provide information for the analysis, use the implemented solution and contribute towards its evaluation. The third party user is likely to be a user (or potential user) of a computer system in business or in other organisations in the community. Whilst a teacher could act as the third party user, this arrangement is far from ideal. Candidates should be encouraged to look beyond school life into businesses and organisations in the community of the surrounding area for their projects; involve the organisation and present them with a report on the work which has been carried out, including an evaluation of this work.

Candidates should make the final choice of context for the project, although the supervisor should give guidance about project suitability. This should include guidance on the appropriateness of an implementation on a stand-alone or networked small computer system or other available facility. In a well organised project the candidate will focus on the production of an overall system analysis and design. The solution may be implemented using one or more of: a programming language, pre-written modules and toolkits, applications software and programmable packages. Brief descriptions of any programming languages or software packages used, together with reasons for their selection, should be included in the report.

Where the solution has involved programming the candidate should:

- annotate listings;
- explain each selection of the program with appropriate algorithm descriptions, which should be language independent;
- define variables by name, type and function where appropriate;
- define clearly, and identify the purpose of, functions subroutines and procedures.

Where the solution has been produced with a software package which has **not** involved programming, candidates should:

- explain each section of the solution with appropriate algorithm descriptions;
- define the purpose and inter-relationship of modules within the system;
- clearly annotate the results produced.

Test data should be devised and used systematically to test the package thoroughly. The choice of test data used, and the reason for choice, should be included. A description of the methods of testing should also be included, together with evidence of testing.

The projects should contain the title, a contents list, a description and justification of investigation, analysis, design and methods used, an evaluation and bibliography. Pages should be clearly numbered. Appropriate evidence of development, testing and implementation, such as screen dumps or photographs of screen layouts and printouts, paper based user documentation and a letter from the third party user to say that the system has been developed satisfactorily, must support the report. Candidates should **not** submit video recordings, magnetic or optical media as supporting evidence but can make references to web pages available over the Internet.

Candidates should choose a well-defined user driven problem of an appropriate size, which enables them to demonstrate their skills in Analysis, Design, Development, Testing, Implementation, Documentation and Evaluation. The project should involve the skills attained by studying the other modules of this specification.

This means that the projects will show more technical skills than projects for Information and Communications Technology, which demonstrate the skills of providing and communicating information, possibly to a wide audience.

The computing projects may involve programming, or at best the tailoring of a generic applications package such as a database, and may involve the choosing and installing of hardware.

Some possible ideas are given below. However, it is important that candidates realise that the project chosen should be user driven. This means that candidates should understand that they should find possible users of computing systems and find out their needs before developing an idea.

The examples do not show the analysis of the problem nor do they give all the facilities, which may be required by the user. For example, the teacher who requires a system to deal with inter-form matches may well want it to include school matches. The examples only give a few ideas about the type of project, which could provide candidates with the opportunities to demonstrate Advanced GCE Computing skills and an understanding of solving real life problems.

Example 1

Miss Jacobs is Head of a P.E. Department in a comprehensive school and has to keep results of inter-form matches and produce league tables. When she receives the results of matches, she has to update the league table. This involves altering the number of games played, won, lost and drawn. She then calculates the total number of points each team has. A win is worth three points and a draw one point. There are no points for a loss. When she has added the results to the league table, she wants it sorted into order of points, the team with the highest number of points first. If two teams have the same number of points, then the team with the best goal difference is first. Goal difference is goals for minus goals against. If two teams are still equal, she places them in the ratio of goals for to goals against. Finally, if the teams are still tied, she places then in alphabetical order.

At present the number of forms in each year is an even number and is usually in the range from 6 to 10. It should be noted that the number of children entering the school is increasing and the number of forms in a year is not necessarily going to be an even number in future. Miss Jacobs wants a computer to print the fixture lists given the number of forms and their codes and to print the league tables.

The computer must produce the fixture list for each tournament and a team can only play once on a given day.

This provides the candidate with a starting point to begin the analysis of the problem. The analysis will reveal sub problems such as the need to produce fixture lists, keep match records...

The candidate must then design a solution to the problem, which meets the agreed criteria.

This problem could be solved by use of application software tailored to the needs of the teacher. Such a solution would require the use of programming to process the results. Alternatively, an argument can be made for writing the code in a high level language. The examiner would expect the candidate to provide a reasoned comparison between the possible methods of solution, justifying their final choice.

Example 2

Ms Jenkins has to be away from home quite a lot and wants a burglar alarm system for her house. She is thinking of buying a computer system to monitor her house and turn lights and motors on and off as situations change. Her specifications for the system are as follows.

It should turn on an alarm if someone enters the house when the alarm is set and also floodlight the house if it is dark. To give the impression of someone being in the house when she is away it should turn lights on (when it is dark) in rooms she has specified before going away. Also she would like to specify which curtains are to be closed when it gets dark. When it becomes light again the curtains should open and the lights should be turned off. She would prefer that these actions should not occur simultaneously and, to a certain extent, randomly.

Each time she starts the system, she wants to be able to decide which of these facilities are to be used. She also wants to be able to decide which rooms have to have lights turned on and off and which curtains are to be opened and closed.

Further, she would like her garage doors to open automatically when she comes home. She doesn’t mind using the car horn during the day to cause the doors to open but is not happy using it at night when people are likely to be in bed. However, she wants a secure system for operating the garage doors.

Ms Jenkins would like you to design a computer system that she could use to satisfy the above requirements. She would like to see a prototype of the design, using something like a doll’s house, before giving her permission for the system to be installed.

The candidate will need to analyse the problem and the hardware available, before coming up with suggestions which can be taken back to Ms Jenkins for approval. In this case the candidate would need to work closely with the user to determine the exact system requirements and suitable areas for computer control.

The examiner would be looking for a suitable control program to be written.

Example 3

Mr Green is a teacher has asked a student to develop an adventure game for his subject and age group of children. The game should allow the user to move from one location to another according to the solution of problems set by the system, or perhaps by collecting objects.

The game should be motivating for his students and educational.

Interest could be increased by inclusion of a score or solving problems against a time limit.

The candidate will need to analyse the type of content required according to subject matter, and age and ability of the children. The design will require a consideration of the user interface, and the implementation will need to be by the writing of code in a high level language.

With projects of this sort it is very easy for the candidate to be too ambitious, and care should be taken by the tutor to ensure the candidate is realistic in their aims.

Guidance on Marking Computing Projects

Computing projects are assessed as follows:

•	Definition and Analysis	**[25 marks]**
•	Design	**[21 marks]**
•	Software Development, Testing and Implementation	**[35 marks]**
•	Documentation	**[24 marks]**
•	Evaluation	**[15 marks]**

(a) Definition, Investigation and Analysis [25 marks]

(i) Definition - nature of the problem solved [5 marks]

A candidate should not expect the examiner to be familiar with the theory and practice in the area of the chosen system. There should be a brief description of the organisation (for example, firm or business) involved and the current methods used in the chosen areas that may form the basis of the project. A clear statement of the origins and form of data should be given. At this stage the exact scope of the project may not be known and it may lead to the arranging of an interview with the user.

1 Vague description of the organisation.

2 Some description of both the stages of study and organisation involved.

3 Good description of either the area or organisation with some description of the other.

4 Clear description with one element missing (for example, origins of data).

5 Excellent description with all elements present.

(ii) Investigation and Analysis [20 marks]

This section is the ‘systems analysis’. The question is not how a system performs detailed tasks, but rather how the project progresses from the original data to the results. The candidate should describe how the user requirements were ascertained (possibly by long discussions with the users; question and answer sessions should be recorded and outcomes agreed). A clear requirements specification should be defined. Alternative outline solutions should be discussed and evaluated against one another.

1 - 5 Some elements have been discussed but little or no user involvement.

6 - 10 Some evidence that an attempt has been made to interview the user and some recording of it has been made. Attempts at some of the other items have been made. An attempt has been made to develop a requirement specification.

11 - 15 Good user involvement and recording of the interview(s). Most of the necessary items have been covered including a detailed discussion of alternative approaches. However, one or two items have been omitted. A requirements specification is present but with some omissions.

16 - 20 Excellent user involvement with detailed recording of the user’s requirements. Alternative approaches have been discussed in depth. All other items must be present, showing a thorough analysis of the system to be computerised. A detailed requirements specification has been produced.

(b) Design [21 marks]

(i) Nature of the solution [13 marks]

A detailed systems design (including diagrams as appropriate) should be produced and agreed with the users. Proposed record, file and data structures should be described and design limitations should be included. Design of data capture forms, input formats (with examples of screen layouts) and output formats should be included here where relevant. A detailed summary of the aims and objectives should also be included. These items are the design specifications, which should be agreed with the user.

1 - 2 Some vague discussion of what the system will do with a brief diagrammatic representation of the new system.

3 - 6 The major objectives of the new system have been adequately summarised, *but omissions have been made*. There is a brief outline of a design specification, including mock ups of inputs and outputs, process model described (including a diagram: structure diagram, data flow diagram or system flowchart). However there is a lack of completeness with omissions from the process model, inputs and outputs. Data structures have been identified but there may be inadequate detail.

7 - 10 A clear set of objectives have been defined and a full design specification is included but there may be some errors or logical inconsistencies, for example validation specified may be inadequate or field lengths incorrect.

11-13 A clear set of objectives with a detailed and complete design specification, which is logically correct. The are also detailed written descriptions of any processes/modules and a clear, complete definition of any data structures. The specification is sufficient for someone to pick up and develop an end result using the software and hardware specified in the requirements specification.

(ii) Intended benefits [3 marks]

There should be some discussion of the relative merits of the intended system and of the previous mode of operation. This may include any degree of generality beyond the original scope of the system.

One mark should be awarded for each valid benefit up to a maximum of three marks.

(iii) Limits of the scope of the solution [5 marks]

This may include volume (sizing limitations) and limitations of the facilities used. For full marks there must be some estimate of the size of the files required for the implemented system.

1 A vague discussion of what the system limitations are.

2 - 3 The major limitations of the system have been adequately summarised, but omissions have been made.

4 - 5 A detailed description of the system limitations has been given, including the estimate of the size of the files required for the implemented system.

(c) Software Development, Testing and Implementation [35 marks]

(i) Development and Testing [18 marks]

A technical description of how the solution relates to the design specification produced and agreed with the user should be included. It is the responsibility of the candidates to produce evidence of their development work and for producing a test plan for the system. It is vital to produce test cases and to show that they work. To do this, it is necessary not only to have test data, but to know what the expected results are with that data.

An attempt should be made to show that all parts of the system have been tested, including those sections dealing with unexpected or invalid data as well as extreme cases. Showing that many other cases of test data are likely to work - by including the outputs that they produce - is another important feature. Evidence of testing is essential. Comments by teachers and others are of value, but the test plan must be supported by evidence in the report of a properly designed testing process. The examiner must be left in no doubt the system actually works in the target environment. This evidence may be in the form of hardcopy output (possibly including screen dumps), photographs or VHS video.

1 - 4 Program listings or evidence of tailoring of a software package is provided in the form of printouts but with no annotation or relationship to a test plan or test run. The developed solution does not fulfil the design specification. A collection of hardcopy test run outputs with no test plan, or a test plan with no hardcopy evidence may also be present. A teacher may award up to 2 marks if they have been shown the system working satisfactorily and there is no hard evidence in the project report.

5 – 8 Program listings or evidence of tailored software packages are provided in the form of printouts. Data structures are illustrated as part of the listings where appropriate, detailing their purpose. There is some annotation evident to illustrate how the package was tailored for a particular purpose or to indicate the purpose of sections of code in a program listing. The developed solution has logical flaws and does not fulfil the design specification. There is little evidence of testing with a badly developed test plan with clear omissions. There is no description of the relationship between the structure of the development work and the testing in evidence.

9 - 13 Program listings or evidence of tailored software packages are provided in the form of printouts. Data structures are illustrated as part of the listings where appropriate, detailing their purpose. There is some annotation evident to illustrate how the package was tailored for a particular purpose or to indicate the purpose of sections of code in a program listing. The developed solution partially fulfils the design specification. There should be at least eight test runs together with a test plan and hardcopy evidence. However, the test plan has omissions in it and/or not all cases have been tested (i.e., have no evidence of testing).

14 - 18 Program listings or evidence of tailored software packages are provided in the form of printouts. Data structures are illustrated as part of the listings where appropriate, detailing their purpose. There is a full set of printouts showing input and output as well as data structures. All hardcopy is fully annotated and cross-referenced. The developed solution completely fulfils the design specification. A full test plan, with evidence of each test run is present in the report, together with the expected output.

The test plan should cover as many different paths through the system as is feasible, including valid, invalid and extreme cases. Marks may be lost for lack of evidence of a particular test run or lack of expected results.

(ii) Implementation [10 marks]

It is recognised that the user organisation (preferably 'third party') may not fully implement the system, although this is the ultimate aim. However, to score any marks in this section there must be some evidence that the person for whom the system was written has seen the system in operation. This can be done in a number of ways: such as by inviting the user to see the product and allowing the candidate to demonstrate the system, or by taking the system to the user involved. There should be an implementation plan written, including details of system changeover, training required and details of user testing.

0 No evidence that the **third party user** has used the system. No written implementation plan.

1 - 4 Details of system changeover have been documented with some recognition that the user(s) will require training. Some evidence of user testing is given, usually by questionnaire or written comments by fellow students or others who were not directly involved in the development of the system.

5 - 7 A good implementation plan with details of training required. There is written evidence available from the third party user indicating that they have seen the system in operation.

8 -10 A clear and detailed implementation plan, including detailed stages of user testing. All aspects of user testing, user acceptance, implementation and system changeover have been documented. There is written evidence available from the user that the system has been fully tested.

(iii) Appropriateness of structure and exploitation of available facilities [7 marks]

Some discussion of the suitability of methods and any product (e.g., hardware or software) used for the particular system should be included. Some recognition and discussion of the problems encountered and actions taken when appropriate should also be included. A log of such problems could be kept. Suitability for subsequent maintainability and extendibility.

1 - 3 Some attempt at discussing either the suitability of the hardware and software, or the problems encountered.

4 - 7 A complete discussion of the hardware and software available and how they were suitable in solving the given problem, together with a good, informative explanation of the problems encountered and how they were overcome.

(d) Documentation [24 marks]

(i) Technical [10 marks]

Much of the documentation will have been produced as a by-product of design and development work and also as part of writing up the report to date. However a technical guide is a standalone document produced to facilitate easy maintenance and upgrade of a system. The contents of the guide should, where relevant, include the following: record, file and data structures used; database modelling and organisation including relationships, screens, reports and menus; data dictionary; data flow (or navigation paths); annotated program listings; detailed flowcharts; details of the algorithms and formulae use. All parts of the guide should be fully annotated since this is very important for subsequent development of the system. The specifications of the hardware and software on which the system can be implemented should be included.

Since the system in the technical guide will differ from one project to another, professional judgement as to what would be necessary for another analyst to maintain and develop the system has to be made.

1 - 2 Some items are present but little annotation.

3 - 6 One or two major omissions, but the rest is fully annotated.

7 - 10 No major omissions, with all parts fully annotated. Marks will be lost for inadequate items of documentation for example, non-specification of hardware on which the system can be implemented. For full marks the guide should be well presented rather than just a collection of items.

(ii) User [14 marks]

Clear guidance, as friendly as possible, should be given to the user for all operations that they would be required to perform. These would include input format with screens displays, print options, back-ups (file integrity routines), security of access to data and a guide to common errors that may occur. (Note the candidate would not be required to copy out large volumes of any underlying software’s user guide, but to produce a non-technical and easy to follow guide for someone with little computer knowledge). Some mention here of the relationship between items of software and the data they deal with may be relevant. The user guide should be well presented with an index and, where necessary, a glossary of the terms used. Alternatively, an electronic guide could be based around hypertext links (screen dumps will be required).

1 - 2 An incomplete, badly produced guide. No screen displays. Some options briefly described but difficult for the user to follow.

3 - 4 A standalone guide has been produced which, though incomplete, contains details of the required input and output and some error conditions. Some screen displays are present in the guide.

5 - 9 All but one or two options fully described (for example, back-up routines not mentioned). In the main the options are easy for the user to follow with screen displays.

10 - 14 A full user guide with all options described well presented (possibly as booklet) with an index and a glossary. No omission of any of the options available (including back-up routines, guide to common errors). Marks may be lost for inadequate descriptions of some options. For full marks, good on-screen help should exist.

(e) Evaluation [15 marks]

(i) Discussion of the degree of success in meeting the original objectives. [6 marks]

This discussion should demonstrate the candidate's ability to evaluate the effectiveness of the completed system. The original objectives stated in requirements specification should be matched to the achievements, taking into account the limitations. User evaluation is also essential and should arise from a questionnaire or, preferably, direct user evaluation. For full marks it is important that the user provides sets of data as they are likely to occur in practice, and that the results arising from such data be given. This data is typical data rather than test data and it may show up faults or problems that the candidate's own test data failed to find.

0 No discussion present.

1 - 3 Some discussion about a number objectives, but some omissions or inadequate explanation of success or failure.

4 - 6 A full discussion, taking each objective mentioned in (b) (i) and explaining the degree of success in meeting them, (indicating where in the project evidence can be found to support this) or reasons why they were not met.

(ii) Evaluate the users' response to the system [5 marks]

It is important that the user is not assumed to be an expert in computer jargon, so some effort must be made to ensure that the system is user-friendly. It will be assumed that the user will have considerable knowledge of the underlying theory of the business being computerised. Clarity of menus, clear on-screen help and easy methods of inputting data are all examples of how the system can be made user-friendly. Here marks are awarded for the degree of satisfaction that the user indicates in the acceptance procedure. Could the system or its results be used? Was the system specification achieved? Do any system faults still exist? The candidate should evaluate the users' response to the final version of the system.

1 Some effort has been made to make the system user-friendly, but the user still has difficulty using the system.

2 - 3 The system is, in the main, user-friendly, but there is room for improvement (e.g., no on-screen help has been provided). The user indicates that the system could be used but there are some faults, which need to be rectified.

4 - 5 A fully user-friendly system has been produced. The user indicates that the system fully meets the specification given in section (a), and there are no known faults in the system.

(iii) Desirable extensions [4 marks]

As a result of completing the system, the candidate should identify the good and bad points of the final system highlighting any limitations and necessary extensions to the system, indicating how the extensions could be carried out.

1 The candidate identifies the obvious good points of the system and possibly some bad points or limitations.

2 The candidate identifies clearly good and bad points and any limitations.

3 The candidate clearly identifies, good and bad points of the system, limitations and possible extensions.

4 The candidate clearly portrays the good and bad points of the system indicating the limitations, possible extensions and how to carry out the extensions.

Appendix C

Applications Areas

The following are some typical applications areas through which it may be appropriate to teach parts of Module 2511. Business Systems applications (see below) are also appropriate for teaching Module 2508.

Real-Time Computing

Typical applications areas should include transport systems such as traffic signals, lifts, conveyors; domestic systems such as washing machines and control systems for heating; security systems.

Questions will demand the application of logical conditions to problems (for example, to switching in a heating system) but mathematical control theory will **not** be needed.

Simulation

Typical examples of simulations might include continental drift, wind-tunnel simulators, queues for service (single server and multi server), weather forecasting, scheduling of industrial processes; other examples could include economic forecasting, vehicular flow in a town, opinion polls and market surveys.

Business Systems

Typical examples of applications which candidates should have some acquaintance with are: payroll, stock control, warehousing, banking, mail order, Electronic Funds Transfer (EFT), smart cards, booking systems, utility billing systems, hospital records, customer accounts, school administration, self-paced teaching systems, multi-media training materials/presentations, library electronic catalogues, voice systems for handling queries, world wide web and Internet, route finders and railway timetables.

Appendix D

Overlap with GNVQ ICT Units

The following grid shows the modules in these specifications which have some overlap in content with Advanced GNVQ in Information and Communications Technology units.

AS/Advanced GCE	Advanced GNVQ ICT
Module 2506	**Unit(s)**
5.1.1	7318
5.1.2	7318
5.1.3	7329
5.1.4	7319, 7327
5.1.5	7318
5.1.6	7337
Module 2507	
5.2.1	7328
5.2.2	7328
5.2.3	7329
Module 2508	
5.3.1	7319, 7328, 7329
5.3.2	7316
5.3.3	7317
5.3.4	7318
5.3.5	7337, 7338
5.3.6	7323
Module 2509	**Unit(s)**
5.4.1	7328
5.4.2	7336
5.4.3	7327
5.4.4	7320
5.4.5	7337

AS/Advanced GCE	Advanced GNVQ ICT
Module 2510	
5.5.1	7319
5.5.2	7317, 7320, 7328
5.5.3	7317, 7320 or others depending on the project
5.5.4	7317, 7320 or others depending on the project
5.5.5	7317, 7320 or others depending on the project
5.5.6	7317, 7320 or others depending on the project
Module 2511	
5.6.1	7316, 7321, 7323
5.6.2	7322
5.6.3	7339
5.6.4	7321, 7337, 7338