SQA Advanced Diploma in Computing: Software Development (International)

GM5A 48

Course Tutor Guide

Publication date: August 2017

Version: 01

Published by the Scottish Qualifications Authority
The Optima Building, 58 Robertson Street, Glasgow G2 8DQ
Lowden, 24 Wester Shawfair, Dalkeith, EH22 1FD

www.sqa.org.uk

Acknowledgement

SQA acknowledges the valuable contribution that Scotland's colleges have made to the development of SQA Advanced Diplomas.

Further information

Call SQA's Customer Contact Centre on 44 (0) 141 500 5030. Alternatively, complete our Centre Feedback Form.

Equality and inclusion

This qualification has been designed to ensure that there are no unnecessary barriers to learning or assessment. The individual needs of learners should be taken into account when planning learning experiences, selecting assessment methods or considering alternative evidence.

Further advice can be found on our website www.sqa.org.uk/assessmentarrangements.

History of changes

It is anticipated that changes will take place during the life of the qualification and this section will record these changes. Centres are advised to check SQA Connect to confirm they are using the up to date qualification structure.

NOTE: Where a Unit is revised by another Unit:

- No new centres may be approved to offer the Unit which has been revised.
- Centres should only enter students for the Unit which has been revised where they are expected to complete the Unit before its finish date.

Version number	Description	Date

Contents

1 Introduction

1.1 Purpose of course Tutor Guide

This Guide is aimed at staff in SQA Approved Centres who are responsible for the SQA Advanced Diploma in Computing: Software Development. It will assist you in the delivery, assessment and internal verification of the SQA Advanced Diploma by providing information on setting up a course, the course framework, centre support with course delivery, teaching plans and guidance on assessment of the Units within the course. It also provides information about Core Skills, Graded Units, Quality Assurance and Verification.

For the purposes of this document learners are referred to in the following terms:

Learners

This term is used to refer to learners in teaching and learning circumstances.

Candidates

This term is used to refer to learners in assessment related matters.

1.2 Introduction to the Award

The award builds on traditional software development techniques but also incorporates development for mobile platforms and development of small scale applications (apps). The award also incorporates skills in data driven web sites which are typically used in e-commerce.

The Units in this award have been selected to give learners the widest range of skills which are in demand by Industry and to allow access to a range of courses at Universities.

2 Setting up the course

Centres will have been approved to offer this course and will have internal processes for liaising with appropriate parties. Liaison with SQA is normally via the centre's SQA co-ordinator.

It is recommended that a Course Team is set-up by the Institution/Department appointed to oversee the delivery of the course. The Course Team should comprise of all tutors teaching on the course and one tutor should be nominated as Course Team Leader.

The Course Team would have the responsibility for overseeing the quality of delivery on the course and to ensure that academic standards are maintained. The Course Team would also be responsible for monitoring student progress and determining the support required for individuals who are not progressing well.

The Course Team should meet to discuss matters relating to course delivery, assessment and internal verification on a regular basis (usually two/three times a year) to ensure that any action points are achieved and that the course is delivered efficiently and effectively. It is good practice to maintain a record of such meetings to ensure that any action points are noted and accomplished. This will benefit centres and their learners by ensuring that all learners achieve their full potential whilst maintaining the appropriate standards.

It is good practice for centres to maintain a library of master folders for each of the Units within the course — these may be stored electronically as long as all relevant staff have access to them as and when required. Electronic files on a secure part of the centre's network/intranet are particularly appropriate if the award is delivered across different campuses allowing all tutors to access the most up to date materials wherever they are located.

The master folders should contain Unit specifications, teaching materials (including details of learning, teaching and assessment plans; and if appropriate, details of any integration across Units of either teaching or assessment), assessment exemplars and re-assessments. This enables new members of staff to access this valuable resource prior to, and during, delivery of the course.

It is good practice for tutors to familiarise themselves with the Units and specific requirements of the assessments prior to the start of the course. Unit specifications set out the statement of standards and evidence required for achieving each Unit, along with guidance on content and assessment. Assessment exemplars provide an instrument of assessment for each Unit and suggested marking scheme. **The assessment exemplars MUST be kept secure at all times.**

All tutors delivering this course have a collective responsibility to ensure that all learners are supported in a manner that meets their individual needs as they progress through the course.

3 The SQA Advanced Diploma Structure

3.1 General SQA Advanced Diploma Qualification Framework

To be awarded an SQA Advanced Diploma, the student is required to achieve **30 SQA Credits** with a mixture of SCQF level 6, 7 and level 8 Units.

Each Unit is assigned a **SQA Credit** value of either 1 or 2. This SQA Credit value is based approximately on 80 hours of study per credit which consists of 40 hours of structured learning and a further 40 hours of student led study to consolidate and reinforce learning.

Each Unit is also assigned a **Scottish Credit and Qualifications Framework** (**SCQF) level and credit point value.** (See below for further details regarding the SCQF).

Each Unit is assigned an agreed number of SCQF credit points. One point represents a notional 10 hours of study by the learner at the identified level.

3.1.1 The Scottish Credit and Qualifications Framework (SCQF)

The SCQF has 12 levels ranging from National 1 at SCQF level 1, up to Doctorate at level 12. The different levels indicate the level of difficulty of a particular qualification and the difference between levels is dependent on factors such as:

- the complexity and depth of knowledge and understanding
- links to associated academic, vocational or professional practice
- the degree of integration, independence and creativity required
- the range and sophistication of application/practice
- the role(s) taken in relation to other learners/workers in carrying out tasks

3.1.2 SCQF level Descriptors

The SCQF level Descriptors outline the general outcomes of learning at SCQF levels under five broad headings:

- Knowledge and understanding (mainly subject based)
- Practice (applied knowledge and understanding)
- Generic cognitive skills (eg evaluation, critical analysis)
- Communication, numeracy and IT skills; and
- Autonomy, accountability and working with others

The Descriptors allow broad comparisons to be made between the outcomes of any learning and allow learners, employers and the public in general to understand the range of skills and learning that should be achieved at each level. SCQF levels are increasingly used in job advertisements to help employers articulate the skills they require for a particular role and to help potential employees to highlight their skills thus ensuring the right person gets the right job.

SQA Advanced Diploma

For SQA Advanced Diploma courses each Unit is also assigned an SCQF level which will be 6, 7 or 8. These levels indicate the degree of difficulty of the work for that Unit.

Recognition of SCQF levels within SQA Units is identified by the last two digits of a Unit code.

Unit codes ending in	SCQF level	Description
46	SCQF level 6	May be included within first year study
47	SCQF level 7	♦ Approximately equivalent to first year of degree level study ♦ May be an introductory Unit to a subject area at SCQF level 8
48	SCQF level 8	♦ Approximately equivalent to second year of degree level study. ♦ May follow on from and develop knowledge from a level 7 Unit.

Tutors involved in the delivery and assessment of Units would find the SCQF level descriptors helpful in determining the appropriate level of difficulty/complexity required.

For example:

Unit	SQA Credit value	Teaching and learning hours	Unit purpose	SCQF points	SCQF level
Developing Software: Introduction (HP1R 47)	1	80	Introductory level Unit introduces software development constructs and basic of programming	8	7
Software Development: Developing Small Standalone Applications (HP2N 47)	2	160	Extends knowledge of programming basics and demonstrates stages in the software development lifecycle and introduces project development	16	7
Software Development: Object Oriented Programming (HP2L 48)	2	160	Further develops pervious knowledge of programming and progresses the student's learning to a higher level by introducing the Object Oriented paradigm	16	8

For further information about SCQF see **http://www.scqf.org.uk**

3.2 Core Skills

The Core Skills are a group of five skills that are key to learning and working in today's world. Employers have identified Core Skills as those that are most likely to be needed in any work environment. This does not mean that every job will need people who are proficient in all five Core Skills but it does mean that every job will require some level of ability in some or all of these skills.

The five Core Skills are: *Communication, Numeracy, Information and Communication Technology (ICT), Problem Solving* and *Working with Others*. Each Core Skill is available at levels 2 to 6 of the Scottish Credit and Qualifications Framework (SCQF). A brief description of each Core Skill is detailed below. A fuller description of each Core Skill at the SCQF levels 2–6 is available on the SQA's website — **www.sqa.org.uk/international.**

3.2.1 Communication

Communication skills underpin almost all personal, social, learning, and working activity. They are essential in clarifying one's own thoughts, in interacting and conversing with others, in expressing thoughts and in conveying information, feeling and opinions.

The Core Skill in *Communication* has two components:

- Oral Communication
- Written Communication

3.2.2 Numeracy

Numeracy skills are necessary for coping with the demands of everyday life, including work and study. People need to be comfortable with numbers, and with graphs, symbols, diagrams and calculators.

The Core Skill in *Numeracy* has two components:

- Using Graphical Information
- Using Number

3.2.3 Information and Communication Technology (ICT)

Information and Communication Technology (ICT) focuses on the ability to use Information Technology (IT) to process information in ways which will be useful in work and in the home — it is not about developing IT specialists.

The Core Skill in *Information and Communication Technology (ICT)* has two components:

- Accessing Information
- Providing/Creating Information

3.2.4 Working with Others

Working with Others develops the skills needed to co-operate with others in learning and working situations to identify and achieve shared goals.

The Core Skill in *Working with Others* has two components:

- Working Co-operatively with Others
- Reviewing Co-operative Contribution

3.2.5 Problem Solving

Problem Solving develops the skills needed for tackling issues and problems in personal, social, vocational and occupational contexts.

The Core Skill in *Problem Solving* has three components:

- Critical Thinking
- Planning and Organising
- Reviewing and Evaluating

All qualifications offered by SQA are evaluated against the Core Skill framework and where opportunity exist to develop and/or achieve a Core Skill (embedded), these have been noted and a table is provided for each SQA Advanced Diploma showing where these skills are embedded or developed.

For further information about the Core Skills in this award see Appendix 1a.

3.3 Assessment

In the SQA Advanced Diploma in Computing: Software Development course the focus is very much on teaching and learning to develop skills within the areas of Computing and in particular Software Development. Assessment is a natural part of learning and should be seen as a method of measuring student progress in a Unit. In the all the SQA Advanced Diploma in Computing Software Development Units assessment amounts to approximately 10% of the overall Unit time allocation.

The reduction in time spent on assessment is an important aim of this award. Assessment in an SQA Advanced Diploma will cover a variety of knowledge and practical skills as well as the more intellectual skills of planning and evaluating. These together with the Core Skills mean that a large number of different methods are employed to ensure that a learner ‘can do what s/he is supposed to do' and 'knows what s/he is supposed to know’.

A large proportion of Units take a ‘project’ approach using the product of a previous assessment, as the foundation of the next and the purpose is to give the candidate a true reflection of how items being studied integrate and relate to industrial practice. Where this is practical, a holistic approach is encouraged to be taken by centres in assessing across a number of Outcomes within Units or across a number of Units. The benefit of such so-called ‘cross-assessment’, if it goes well, is the achievement of several Outcomes on several Units with just one assessment instrument. A matching disadvantage is that a failure results in several Units not being achieved. It would be wise for centres to consider separating out the ‘retake’ assessments of learners who have failed in their first attempt at a composite assessment instrument. It may be possible to combine the delivery of Units in such a way as to create a thematic delivery of the component Units. The ways in which Units may be integrated is left to centres but thematic delivery, as opposed to discrete Unit delivery, may reduce assessment and improve coherence of content.

The normal rules of re-assessment apply to this award. Candidates are normally permitted one re-assessment, or, in exceptional circumstances, two re-assessments at the discretion of the centre.

3.4 Graded Units

In the framework of every SQA Advanced Diploma there are 3 SQA Credits of Graded Unit. There are Graded Units built into the framework towards the latter part of Year 1 and Year 2. Year 1 Graded Unit is a single credit exam. Year 2 Graded Unit is a 2-SQA-Credit project.

The purpose of Graded Units are to:

- demonstrate that the student has achieved the principal aims of the Group Award
- demonstrate the candidate's ability to integrate the knowledge and understanding gained from other Units making up the Group Award
- grade student performance

Graded Units are specific to the Group Award being delivered, and reflect the principal aims of that Group Award. This means that they will also clearly reflect the uniqueness of the title of the Group Award.

The allocation of grades awarded are as follows:

- Grade A — Student has achieved a mark of 70% or above
- Grade B — Student has achieved a mark of between 60% and 69%
- Grade C — Student has achieved a mark of between 50% and 59%
- F (Fail) — Student has failed to reach the required standard and achieved a mark less than 50%

For details of Graded Units in this award see Section 8.3.

4 SQA Advanced Diploma in Computing: Software Development

4.1 Target audience

The SQA Advanced Diploma in Computing: Software Development develops skills and knowledge in a range of systems analysis and design, development and implementation of a range of software development environments and database analysis and design.

The SQA Advanced Diploma in Computing: Software Development is designed to lead to employment as systems analyst, software designer/developer in applications development, database design and development and web design incorporating e-commerce.

Successful learners should be able to progress to a range of degrees.

4.2 Access to the course

As with all SQA qualifications, access will be at the discretion of the centre and the following recommendations are for guidance only. It should be noted that this qualification will be taught and assessed in English.

Some examples of appropriate formal entry qualifications are specified below. They are not exhaustive or mutually exclusive and may be offered in a variety of combinations.

- An appropriate level of skill in the English language.
- An appropriate level of skill in Mathematics.
- Experience in the use of IT applications software.
- Different combinations of relevant National Qualifications, Vocational Qualifications and equivalent qualifications from other awarding bodies may be acceptable, as would suitable industry standard qualifications at an appropriate level.

4.3 Aims of course

The SQA Advanced Diploma in Computing: Software Development award has a range of aims relating to academic and vocational progression.

The general aims of this award are to:

- Develop candidates' knowledge and skills in planning, developing and evaluating
- Develop employment skills, particularly relating to the IT industry
- Learning and transferable skills (including Core Skills)
- Develop study and research skills
- Develop Provide academic stimulus
- Develop English language skills
- To enable learners to achieve an International award which is studied and assessed in English

The specific aims of this award are to:

- Develop a range of contemporary software development skills that are required by industry
- Prepare learners for employment in competitive international markets
- Prepare learners for progression to degree study within the United Kingdom
- Provide underpinning knowledge associated with appropriate vendor qualifications

4.4 SQA Advanced Diploma in Computing: Software Development Framework

The table below shows the whole framework of Units for the SQA Advanced Diploma in Computing: Software Development and includes their SQA Credit value and SCQF level.

Unit name	Unit code	SQA Credit value	SCQF level
YEAR 1			
Developing Software: Introduction	HP1R 47	1	7
Professionalism and Ethics in Computing	HP29 47	1	7
Computer Systems Fundamentals	HP1T 47	1	7
Troubleshooting Computer Problems	HP1V 47	1	7
Database Design Fundamentals	HP2G 47	1	7
Relational Database Management Systems	HP2J 48	2	8
SQL: Introduction	HP2E 47	1	7
Software Development: Developing Small Standalone Applications	HP2N 47	2	7
Software Development: Programming Foundations	HP2P 47	1	7
Computing: Introduction to Project Management	HP21 47	1	7
Mathematics for Computing 1	HP1H 47	1	7
Team Working in Computing	HP1X 47	1	7
Computing: Graded Unit 1	HP2A 47	1	7

Unit name	Unit code	SQA Credit value	SCQF level
YEAR 2			
Software Development: Object Oriented Programming	HP2L 48	2	8
Systems Development: Object Oriented Analysis and Design	HP2M 48	2	8
Software Development: Data Structures	HP2K 48	2	8
Scripting for Interactivity	HP2D 48	2	8
Self Describing Data (XML)	HP2H 48	1	8
Software Development: Rapid Applications Development and Prototyping	HP2F 48	2	8
Web Development: Dynamically Generated Content	HP2T 48	2	8
Computing: Software Development: Graded Unit 2	HP2R 48	2	8

4.5 Core Skills

All of the revised Units within this award have been assessed and validated against the Core Skills 2013 framework.[1]

Successful learners will exit from the SQA Advanced Diploma in Computing: Software Development with the following Core Skills profile:

Core Skill	Certificated exit level
Communication	SCQF level 5
Numeracy	SCQF level 5
Information and Communication Technology (ICT)	SCQF level 6
Problem Solving	SCQF level 6
Working with Others	SCQF level 6

A detailed analysis of the Core Skills profile is provided in Appendix 1a.

4.6 Graded Unit

Learners will take a 1-SQA-Credit Graded Unit at SCQF level 7 in the first year of the SQA Advanced Diploma Group Award, and a further 2-SQA-Credit Graded Unit at SCQF level 8 in the second year of the SQA Advanced Diploma Group Award.

The Graded Units take the form of:

Computing: Graded Unit 1 (HP2A 47) Examination at SCQF level 7
— 1 SQA Credit

Computing: Software Development Graded Unit 2 (HP2R 48) Project at SCQF level 8
— 2 SQA Credits

Further details are provided in Section 8.3.

[1] Core Skills Framework: an introduction (SQA, Glasgow, 2013)

5 Course delivery of an SQA Advanced Diploma

5.1 How the course is delivered

All tutors must ensure that they deliver this course using teaching methods that engage learners in 'active learning' to encourage them to participate in the learning activities set. All SQA qualifications are designed to enable learners to develop their knowledge and skills and then they are required to apply this new knowledge/skill to a new situation. Criterion-referenced assessments assume that all parties are fully informed of the criteria that learners must achieve and the assessment conditions under which the candidates carry out the assessment activity.[2]

To ensure that learners are fully prepared it is essential that tutors provide as many opportunities as possible for learners to be actively engaged throughout the learning process. Learners should:

- be fully informed of the criteria they must achieve
- be offered a range of learning activities to research, analyse and apply new knowledge/skills to new situations
- be offered opportunity to experience the type of activity that they will be required to carry out as part of the summative assessment
- be able to critically evaluate their personal contribution and to receive feedback from the tutor on how to enhance their understanding

Tutors should develop a learning, teaching and assessment plan for each Unit within the course and provide activities that learners should undertake.

[2] For further information about different assessment activities − whether they be for formative or summative purposes, tutors may wish to complete the online course: Produce HN Assessments for successful prior verification OR read the SQA Guide to Assessment

Each Unit should have a master folder containing the Unit specification, teaching materials, the teaching and assessment plan along with assessment exemplars and re-assessments. The teaching materials and teaching plan should provide details of activities that learners should undertake. Typically they include activities such as small group/whole class discussion, group problem solving, eg analysing a project brief and offering solutions based on the new learning, group project work to find examples, to research new knowledge and to present their findings to their fellow learners.

The following is a list of learning activities but it is not exhaustive:

- Lectures
- Tutorials
- Study packs
- Problem based scenarios
- Case studies
- Group/team work
- Online materials
- IT based teaching materials
- Projects
- Quizzes
- Research and presentation of findings to fellow learners
- Role play
- Short-response questions, multiple-choice questions
- Create questions for other learners (with answers), etc

Tutors should consider the nature of the assessment method as well as the assessment content when planning learning activities so that learners are appropriately prepared.

It is the tutor's responsibility to explain to the learners what is required of them and then to direct, encourage, co-ordinate and support learners to complete the activity. It is also the tutor's responsibility to ensure the resources needed are available to the learners. Materials should be reviewed on a regular basis to ensure they are still relevant.

Some activities could require learners to work in pairs or small groups to discuss issues or to solve a given problem. Other activities could require the student to undertake some independent research out with the classroom and to bring their findings to the next lesson and present this to the class in a report or presentation format. Some Units will require the student to undertake independent reading and learners should be prepared to discuss key issues within the classroom as organised and led by the tutor.

In practical skills classes, learners should be directed to use practice exercises to enable them to become proficient. Tutors may demonstrate the skill first and then coach the learners individually when unsure. In terms of developing independent learners, in the case of information technology, learners should be encouraged to independently use the online Help facilities within applications. It should be noted that even in practical classes, learners should be encouraged to work in small groups and to support one another as part of the learning process — by explaining to another, a student has to reformulate and communicate the learning point thus deepening their learning.

When undertaking group work, learners should be encouraged/directed to work with different groups each time they attempt a new task so that they get to know and work with a wide range of individuals. The groups should be given clear task activities. Tutors should note the various roles assigned to the group members and they should set a time limit for the completion of the task.

At the end of each activity tutors should make time to receive feedback from each group so that they can assess knowledge and understanding and use the feedback session to repeat important key points and to clear up any misunderstandings. Tutors must also provide feedback to learners on their performance in activities, etc.

Where centres have access to electronic resources such as Virtual Learning Environments, Blogs, Wikis, etc — tutors/lecturers are encouraged to use these collaborative tools in the learning process.

5.2 Support for learners

All tutors delivering on a course have a collective responsibility to ensure that all learners are supported in a manner that meets their individual needs as they progress through the course.

Each individual tutor has a role to monitor an individual student's understanding and progress at Unit level and feed comments to the Course Team. At individual Unit level, tutors may wish to use a range of mechanisms to support learners and to establish if learners are progressing well on the course.

6 SQA Advanced Diploma in Computing: Software Development course delivery

6.1 Teaching plan

The Units that make up this Group Award are listed in Section 6.2 — ***Overview of Units.***

This Section — ***Teaching plan*** — provides a ***suggested*** delivery schedule for the Units and highlights the best way to sequence the Units over two years.

When constructing this teaching plan consideration was given to the following points:

Year 1 will contain Units which are mainly at SCQF level 7 with a small number of level 8 Units.

Year 2 will contain Units which are mainly at SCQF level 8 and there should be a natural progression from some of the Units delivered in Year 1 to those being delivered in Year 2.

- Some Units are 2-SQA-Credit and thought must be given as to whether the Unit should be covered in one Semester or across the whole year. This is particularly the case with Year 2 and suggested Section 6.1.2.
- Finally, the Graded Units completed at the end of Years 1 and 2 are based on some of the mandatory Units. The Units being assessed as part of the Graded Unit, must be delivered and assessed to ensure that sufficient learning will have taken place to enable the learners a fair opportunity at achieving the Graded Unit at an appropriate grade.

Learners will study 15-credits worth of Units in each year. One Semester will have 7 credits and one Semester 8 credits. The weighting of these Units will depend on the relationship of the relevant Unit with other Units in terms of prior knowledge needed and/or complementary knowledge.

Rationale for the suggested delivery schedule

Two suggested delivery schedules have been given for Year 1 and Year 2 of SQA Advanced Diploma in Computing: Software Development — Option A and Option B.

The majority of the SCQF level 7 Units have been included in Year 1, so that learners are provided with introductory knowledge and understanding in the subject areas of:

- Software development and programming
- Computer systems
- Team working
- Troubleshooting Computer Problems
- Databases including SQL and relational database
- Project management

The skills gained in these subjects will provide underpinning knowledge for study in Year 2.

The majority of the Units in Year 2 are SCQF level 8 Units that will develop on the knowledge and understanding gained in Year 1. Learners will be able to extend their knowledge and skills in:

- Object oriented design and programming
- Data structures
- Scripting
- Prototype development
- Data usage in a web environment

6.1.1 Year 1: Suggested delivery schedule

Year 1 − Option A

Semester 1			
Unit code	**Unit title**	**SQA Credit value**	**SCQF level**
HP1R 47	Developing Software: Introduction	1	7
HP29 47	Professionalism and Ethics in Computing	1	7
HP1T 47	Computer Systems Fundamentals	1	7
HP2G 47	Database Design Fundamentals	1	7
HP2P 47	Software Development: Programming Foundations	1	7
HP21 47	Computing: Introduction to Project Management	1	7
HP1V 47	Troubleshooting Computer Problems	1	7

Semester 2			
Unit code	**Unit title**	**SQA Credit value**	**SCQF level**
HP1X 47	Team Working in Computing	1	7
HP2A 47	Computing: Graded Unit 1	1	7
HP2J 48	Relational Database Management Systems	2	8
HP2N 47	Software Development: Developing Small Standalone Applications	2	7
HP2E 47	SQL: Introduction	1	7
HP1H 47	Mathematics for Computing 1	1	7

Summary of Year 1 Option A

Semester	Total number of Units	Number of SQA Credits at		SCQF credit points		Number of Units completed	Number of Units partially completed
		SCQF level 7	SCQF level 8	SCQF level 7	SCQF level 8		
1	7	7	0	56	0	7	0
2	6	6	2	48	16	6	0

Year 1 — Option A — Semester overview

Semester 1	Semester 2
Developing Software: Introduction (HP1R 47) (1 SQA Credit)	Computing: Graded Unit 1 (HP2A 47) (1 SQA Credit)
Professionalism and Ethics in Computing (HP29 47) (1 SQA Credit)	Software Development: Developing Small Standalone Applications (HP2N 47) (2 SQA Credits)
Computer Systems Fundamentals (HP1T 47) (1 SQA Credit)	
Database Design Fundamentals (HP2G 47) (1 SQA Credit)	Relational Database Management Systems (HP2J 48) (2 SQA Credits)
Software Development: Programming Foundations (HP2P 47) (1 SQA Credit)	
Computing: Introduction to Project Management (HP21 47) (1 SQA Credit)	SQL: Introduction (HP2E 47) (1 SQA Credit)
Troubleshooting Computer Problems (HP1V 47) (1 SQA Credit)	Mathematics for Computing 1 (HP1H 47) (1 SQA Credit)
	Team Working in Computing (HP1X 47) (1 SQA Credit)

Rationale for Year 1 (Option A)

Option 1 offers a range of introductory Units in Semester 1 which will allow learners to develop foundation knowledge for Units in Semester 2 and Graded Unit 1.

Developing Software: Introduction (HP1R 47) is a key foundation for the SQA Advanced Diploma in Computing: Software Development. It is therefore essential that this be delivered at an early stage. To enhance and further develop skills this can be delivered in conjunction with *Software Development: Programming Foundations* (HP2P 47).

The Units which form the basis of the Graded Unit exam are generic to all SQA Advanced Diplomas within Computing and are designed to develop generic Computing skills and understanding. Introduction of these Units at an early stage will allow learners to build on their earning in these Units and assist in development of a good understanding of further Units of study. These Units are:

HP1T 47 *Computer Systems Fundamentals*
HP29 47 *Professionalism and Ethics in Computing*
HP1V 47 *Troubleshooting Computer Problems*
HP1R 47 *Developing Software: Introduction*

Computing: Introduction to Project Management (HP21 47) will allow learners to develop an understanding of prioritising and time management. This Unit should be delivered prior to the *Team Working in Computing* (HP1X 47) as this requires learners to manage a group project.

In Semester 2 *Software Development: Developing Small Standalone Applications* (HP2N 47) will further develop skills from Semester 1 and extend the foundation for Year 2 study.

Key features of Year 1 (Option A)

- All Units are delivered in a single Semester
- Underpinning knowledge is gained in Semester 1 and will allow learners to gain confidence with a sound foundation on which to base further learning
- All subjects for Graded Unit 1 are delivered in Semester 1
- Project Management skills are gained in Semester 1. These will be required in *Team Working in Computing* (HP1X 47)
- Semester 2 would allow integration of *Relational Database Management Systems* (HP2J 48) and *SQL: Introduction* (HP2E 47) for delivery through the Oracle Academy programme
- Allows monitoring of progress at the end of Semester 1

Year 1 — Option B

Semester 2			
Unit code	**Unit title**	**SQA Credit value**	**SCQF level**
HP1R 47	Developing Software: Introduction	1	7
HP1T 47	Computer Systems Fundamentals	1	7
HP21 47	Computing: Introduction to Project Management	1	7
HP1H 47	Mathematics for Computing 1	1	7
HP2A 47	Computing: Graded Unit 1 (Exam)	1	7
HP29 47	Professionalism and Ethics in Computing	1	7
HP2G 47	Database Design Fundamentals	1	7
HP1V 47	Troubleshooting Computer Problems	1	7

Semester 2			
Unit code	**Unit title**	**SQA Credit value**	**SCQF level**
HP1X 47	Team Working in Computing	1	7
HP2P 47	Software Development: Programming Foundations	1	7
HP2E 47	SQL: Introduction	1	7
HP2N 47	Software Development: Developing Small Standalone Applications	2	7
HP2J 48	Relational Database Management Systems	2	8

Summary of Year 1 Option B

Semester	**Total number of Units**	**Number of SQA Credits at**		**SCQF credit points**		**Number of Units completed**	**Number of Units partially completed**
		SCQF level 7	SCQF level 8	SCQF level 7	SCQF level 8		
1	8	8	0	64	0	8	0
2	5	5	2	40	16	5	0

Year 1 — Option B — Semester overview

Semester 1	**Semester 2**
Developing Software: Introduction (HP1R 47) (1 SQA Credit)	Software Development: Programming Foundations (HP2P 47) (1 SQA Credit)
Professionalism and Ethics in Computing (HP29 47) (1 SQA Credit)	Software Development: Developing Small Standalone Applications (HP2N 47) (2 SQA Credits)
Computer Systems Fundamentals (HP1T 47) (1 SQA Credit)	
Database Design Fundamentals (HP2G 47) (1 SQA Credit)	Relational Database Management Systems (HP2J 48) (2 SQA Credits)
Computing: Graded Unit 1 (HP21 47) (1 SQA Credit)	
Computing: Introduction to Project Management (HP21 47) (1 SQA Credit)	SQL: Introduction (HP2E 47) (1 SQA Credit)
Troubleshooting Computer Problems (HP1V 47) (1 SQA Credit)	Team Working in Computing (HP1X 47) (1 SQA Credit)
Mathematics for Computing 1 (HP1H 47) (1 SQA Credit)	

Rationale for Year 1 (Option B)

Option B introduces all foundation Units and the Graded Unit 1 examination in Semester 1. This will allow tutors to integrate subject teaching with exam techniques and will concentrate the focus of the exam in Semester 1. It would be expected that the actual examination would be scheduled towards the end of Semester 1.

Semester 2 would then allow the development of practical skills which will form the basis of the Year 2. This would also allow a more relaxed development of practical skills as the prospect of the formal assessment is removed.

Key features of Year 1 (Option B)

- All Units are delivered in a single Semester
- All subjects for Graded Unit 1 are delivered in Semester 1
- Graded Unit scheduled in Semester 1
- Underpinning knowledge is gained in Semester
- Project management skills are gained in Semester 1, These will be required in *Team Working in Computing* (HP1X 47)
- Semester 2 would allow focus on development of practical skills
- Allows monitoring of progress at the end of Semester 1

6.1.2 Year 2: Suggested delivery schedule

Year 2 — Option A

Year 2 contains a large number of 2-SQA-Credit Units. Centres may decide to extend the Units over both Semesters, as in Option B, giving an extended period of learning. Option A has no subjects spanning both Semesters but would require more classes per week per subject.

Semester 1			
Unit code	**Unit title**	**SQA Credit value**	**SCQF level**
HP2L 48	Software Development: Object Oriented Programming	2	8
HP2M 48	Systems Development: Object Oriented Analysis and Design	2	8
HP2K 48	Software Development: Data Structures	2	8
HP2F 48	Software Development: Rapid Applications Development and Prototyping	2	8

Semester 2			
Unit code	**Unit title**	**SQA Credit value**	**SCQF level**
HP2D 48	Scripting for Interactivity	2	8
HP2H 48	Self Describing Data (XML)	1	8
HP2T 48	Web Development: Dynamically Generated Content	2	8
HP2R 48	Computing: Software Development: Graded Unit 2	2	8

Summary of Year 2 Option A

Semester	Total number of Units	Number of SQA Credits at		SCQF credit points		Number of Units completed	Number of Units partially completed
		SCQF level 7	SCQF level 8	SCQF level 7	SCQF level 8		
1	4	0	8	0	64	4	0
2	4	0	7	0	56	4	0

Year 2 — Option A — Semester overview

Semester 1	Semester 2
Software Development: Object Oriented Programming (HP2L 48) (2 SQA Credits)	Scripting for Interactivity (HP2D 48) (2 SQA Credits)
Systems Development: Object Oriented Analysis and Design (HP2M 48) (2 SQA Credits)	Web Development: Dynamically Generated Content (HP2T 48) (2 SQA Credits)
Software Development: Data Structures (HP2K 48 (2 SQA Credits)	Computing: Software Development: Graded Unit 2 (HP2R 48) (2 SQA Credits)
Software Development: Rapid Applications Development and Prototyping (HP2F 48) (2 SQA Credits)	Self Describing Data (XML) (HP2H 48) (1 SQA Credit)

Rationale for Year 2 (Option A)

Option A has no Units extending over the two Semesters. This means that teaching and delivery will be intense over each of the 17 weeks, thus allowing focus on a small number of subjects in each Semester. It also ensures that learners have covered the likely component Units for Graded Unit 2 prior to starting the project. This also means that assessment is more evenly distributed throughout the academic session.

Option A has four subjects studied over each of the two Semesters. Semester 1 is four 2-SQA-Credit Units. This means that the remaining four subjects comprise of three 2-SQA-Credit Units and a 1-SQA-Credit Unit. This minimises the amount of assessment towards the end of the course and would allow more focus at that time on the Graded Unit project.

Key features of Year 2 (Option A)

- All Units are delivered in a single Semester
- All subjects for Graded Unit 2 are delivered in Semester 1
- Spreads the assessments throughout the academic year
- Reduces the amount of assessment at the end of the course
- Allows monitoring of progress at the end of Semester 1

Year 2 − Option B

Semester 1			
Unit code	**Unit title**	**Credit value**	**SCQF level**
HP2L 48	Software Development: Object Oriented Programming	2	8
HP2M 48	Systems Development: Object Oriented Analysis and Design	2	8
HP2H 48	Self Describing Data (XML)	1	8
HP2K 48	Software Development: Data Structures	2	8
HP2F 48	Software Development: Rapid Applications Development and Prototyping	2	8
HP2R 48	Computing: Software Development: Graded Unit 2	2	8

Semester 2			
Unit code	**Unit title**	**Credit value**	**SCQF level**
**HP2K 48	Software Development: Data Structures	2	8
**HP2F 48	Software Development: Rapid Applications Development and Prototyping	2	8
**HP2R 48	Computing: Software Development: Graded Unit 2 (Project)	2	8
HP2D 48	Scripting for Interactivity	2	8
HP2T 48	Web Development: Dynamically Generated Content	2	8

** Units delivered over two Semesters

Summary of Year 2 Option B

Semester	**Total number of Units**	**Number of SQA Credits at**		**SCQF credit points**		**Number of Units completed**	**Number of Units partially completed**
		SCQF level 7	SCQF level 8	SCQF level 7	SCQF level 8		
1	6	0	11	0	88*	3	3
2	5	0	10	0	80	5	0

Year 2 — Option B — Semester overview

<table>
<tr><th>Semester 1</th><th>Semester 2</th></tr>
<tr><td>Software Development: Object Oriented Programming (HP2L 48) (2 SQA Credits)</td><td>Scripting for Interactivity (HP2D 48) (2 SQA Credits)</td></tr>
<tr><td>Systems Development: Object Oriented Analysis and Design (HP2M 48)
(2 SQA Credits)</td><td>Web Development: Dynamically Generated Content (HP2T 48) (2 SQA Credits)</td></tr>
<tr><td colspan="2">Software Development: Data Structures (HP2K 48) (2 SQA Credits)</td></tr>
<tr><td colspan="2">Software Development: Rapid Applications Development and Prototyping (HP2F 48)
(2 SQA Credits)</td></tr>
<tr><td>Self Describing Data (XML) (HP2H 48)
(1 SQA Credit)</td><td></td></tr>
<tr><td colspan="2">Computing: Software Development: Graded Unit 2 (HP2R 48) (2 SQA Credits)</td></tr>
</table>

Rationale for Year 2 (Option B)

Option B has some Units which span both Semesters. This can be beneficial as it gives learners a physically longer period of learning although the number of actual hours will remain the same.

The two main Units likely to be included in the Graded Unit project *(Software Development: Object Oriented Programming* (HP2L 48) and *Systems Development: Object Oriented Analysis and Design* (HP2M 48)) are both completed in Semester 1. The introduction of Graded Unit 2 in Semester 2 will allow learners to begin work on the planning stage and initial analysis. By the time they are ready to develop they should have sufficient knowledge to enable them to what is required.

The delivery of *Scripting for Interactivity* (HP2D 48) and *Web Development: Dynamically Generated Content* (HP2T 48) in the same Semester may allow some integration of these Units.

Key features of Year 2 (Option A)

- Some Units are delivered over the entire academic year
- Longer period of time (34 weeks) for Graded Unit 2 although same number of hours
- Core subjects for Graded Unit 2 are delivered in Semester 1
- Spreads the assessments throughout the academic year
- Allows some monitoring of progress at the end of Semester 1

6.2 Overview of Units

An overview of each Unit delivered in Years 1 and 2 is given below. However, tutors should refer to the Unit specification for full details of the Knowledge and/or Skills to be covered and Evidence Requirements. The Evidence Requirements clearly state the type of evidence required, the standard of evidence required and any conditions of assessment. The Unit specification also contains guidance on the delivery and assessment of the Unit.

Year 1 Units

Developing Software: Introduction — HP1R 47

This 1-SQA-Credit Unit is designed to enable learners to develop basic software development skills. The design and implementation of the constructs of programming (variables, sequence, selection, iteration, functions and parameter passing) will be covered in the context of a development environment. Test plans, test cases and program documentation will also be introduced.

This introduction will provide a basis for further study in software development using a range of programming languages.

There are two Outcomes in this Unit:

1 Implement and test code to carry out tasks following a given design.
2 Prepare technical documentation in line with good practice.

Throughout this Unit learners will learn to use tools and techniques for basic software development using a suitable development environment determined such as structured C++ or any other language with a structured development environment. The choice of language will be at the discretion the centre.

Learners will learn how to code simple tasks and how this code interacts with the system. In addition they will learn to troubleshoot your code so that it runs error free and produces the desired results. This will involve rigorous testing and it is necessary that tutors stress the importance of testing and techniques that can be used and allow learners to develop skills in testing.

Learners should be given designs to follow and this should be used to help code solutions.

This Unit is assessed by practical assessment. All theoretical aspects of development should be demonstrated within a practical context.

Content of this Unit is also assessed in Graded Unit 1.

Professionalism and Ethics in Computing — HP29 47

This 1-SQA-Credit Unit is designed to allow learners to gain the knowledge and understanding required to carry out the day to day duties and activities required of a computing professional in an ethical manner with due attention to business, society and legal requirements.

There are four Outcomes in this Unit:

1 Describe professional bodies relevant to the computing profession.
2 Apply principles of codes of conduct relevant to the computing profession.
3 Describe contemporary legislative concerns for computing professionals.
4 Evaluate ethical considerations in a relevant vocational context.

The Unit consists of four Outcomes which inter-relate to one another, to assist in development of a knowledge base and understanding of a computing professional's responsibilities in regard to:

- Professionalism in duties carried out within job functions including: the advantages of interacting with professional bodies in computing, the need for Continuous Professional Development, awareness of appropriate/acceptable conduct.
- Contemporary legislative requirements.
- Adherence to appropriate ethical conduct.
- Understanding of potential for and resolution of ethical conflict.

The knowledge gained through researching the areas included in the Unit will enable learners to approach future job roles in the computing profession in a responsible and ethical way.

Successful completion of the Unit will be achieved by submission of a single assignment based on questions related to a case study/scenario of a real business situation.

The case study which will form the basis of the assignment should be available to learners early in the Unit scheduling and as a tutor you should allow learners to ask questions to clarify their understanding of the details of the case study/scenario issued.

Content of this Unit is also assessed in Graded Unit 1.

Computer Systems Fundamentals — HP1T 47

This 1-SQA-Credit Unit is designed to provide learners with knowledge about the fundamentals of computer systems and focuses on how the various software and hardware elements interact. The Unit has three main areas, the physical and software elements of a computer system, the number systems and logic used within a computer system and the installation of various types of software. The first two areas are theoretical and the third area is practical.

There are three Outcomes in this Unit:

1 Explain the purpose of the elements of a computer system.
2 Manipulate and explain the uses of number and logic systems used in a computer.
3 Install a range of computer software.

Outcome 1 develops knowledge on the ways in which the central processing Unit communicates with memory and input/output devices. Communication channels such as busses will also be covered. This also covers the layers of the operating system. Computer memory can be of many types and you will learn to distinguish the features of different types of memory. Computer software will cover different types of operating systems, application and security software as well as file systems and structures. Some of the many ways to secure a computer system such as rights, permissions and security software will be introduced. Some of the information will be presented to you in graphical form so that learners can learn to interpret information from, for example, manufacturers' documentation.

The main focus of Outcome 2 is in the application of number systems and logic. This introduces binary and hexadecimal number systems and operations such as add and subtract in these systems and to convert between these systems and decimal numbers. This also covers computer logic and the applications in which it can be used such as using masks in calculating network addresses and changing the case of letters.

Outcome 3 focuses on the practical tasks of installing software. Learners will have the opportunity to apply the theory that they have learned in Outcome 1 to install operating system, application and security software.

Content of this Unit is also assessed in Graded Unit 1.

Troubleshooting Computing Problems — HP1V 47

This 1-SQA-Credit Unit is designed to provide the skills required to develop a possible solution to a computing problem in the context of computer networking or software development Topics covered will cover how to investigate the problem, plan and implement a solution, test and amend it until the problem is resolved and document each step taken to solve the problem.

There are three Outcomes in this Unit:

1 Investigate a computing problem.
2 Plan and implement a solution to a computing problem.
3 Document the steps taken to resolve a computing problem.

In the first Outcome learners will learn how to investigate a computing problem. This will cover approaches to problem solving, test strategies and techniques and problem solving tools and techniques.

In the second Outcome learners will learn how to plan and implement a solution to a computing problem. This will cover planning a solution, implementing a solution and testing the solution.

In the third Outcome learners will learn how to document the steps taken to resolve a computing problem. This will include documenting the investigation, documenting the solution and documenting the testing.

For assessment purposes learners will be presented with a problem to investigate. The problem will be drawn from one or more of the following areas: networking or software development. Learners should develop skills in troubleshooting naturally occurring problems that they may encounter during their course. Learners should be shown how to approach the diagnosis in a logical manner and completed relevant documentation. Documentation could take the form of error logs, test logs or any other appropriate form.

Content of this Unit is also assessed in Graded Unit 1.

Database Units

The following three Units can be integrated and delivered in sequence. They all relate to Relational Database Movement Systems (RDBMS). RDBMS remain crucially important within software development and skills within this part of the industry are constantly in high demand.

The database Units *SQL: Introduction* (HP2E 47) and *Relational Database Management Systems* (HP2J 48) are mapped to the Oracle Academy program These Units can be delivered in conjunction with the Oracle course Database Design and Database Programming with SQL. On successful completion of this oracle course candidates can then be given credit for the Units *SQL: Introduction* (HP2E 47) and *Relational Database Management Systems* (HP2J 48).

Database Design Fundamentals — HP2G 47

This 1-SQA-Credit Unit is the first introduction to relational databases within the award. This will introduce normalised data sets and simple data interrogation techniques.

Learners will acquire the knowledge and learn the skills necessary to enable them to create a normalised RDBMS structure from first principles. To achieve this, Learners will require access to a workstation and also to a suitable SQL-based Relational Database Management System applications package.

There are three Outcomes in this Unit:

1 Create a normalised relational database structure.
2 Write SQL select statements to maintain and update a database structure.
3 To interrogate the database and manipulate the data.

This Unit will involve two major areas of learning. Firstly, breaking down the problem into identifiable and manageable steps from which will be developed the required database structure.

Secondly, the syntax, functions and structures of SQL scripts, with which the learners will use to implement the solutions.

Understanding and development of these skills and techniques will be reinforced throughout with practical exercises. Using a bank of test data, learners will test their scripts to ensure their correct working to meet the user's needs. Learners will be required to amend any errors in their solutions in order to achieve robust, reliable and efficient scripts.

Relational Database Management Systems — HP2J 48

This 2-SQA-Credit Unit will further develop the knowledge and skills gained in the Unit *Database Design Fundamentals* (HP2G 47). Learners will be required to design and create a Relational Database Management System (RDBMS).

This Unit is intended to give an insight into the design and development of these systems. Team working can also be introduced in this Unit and it also provides an opportunity to work in a software development team to achieve set tasks and goals.

There are four Outcomes in this Unit:

1 Identify and use the processes and terminology used in designing a RDBMS.
2 Design an RDBMS from a given scenario.
3 Map the design model to the physical model.
4 Create and run SQL statements/ queries on a RDBMS.

The four Outcomes are assessed by two assessments. A practical assessment covering Outcomes 2, 3 and 4 and a set of questions covering Outcome 1.

SQL: Introduction — HP2E 47

This 1-SQA-Credit Unit further develops design skills gained in the previous database Units but develops the skills necessary to enable learners to create programming scripts using a SQL. Learners will require knowledge of RDBMS and have access to a suitable SQL-based Relational Database Management System applications package.

There are three Outcomes in this Unit:

1 Create and maintain a data storage system.
2 Manipulate data stored within a table structure.
3 Produce formatted reports.

The three Outcomes involve two major areas of learning. Firstly, breaking down the problem into identifiable and manageable steps from which learners will develop the required query script.

Secondly, the learning and development of the syntax, functions and structures of SQL scripts, with which will be used to implement the solutions. Understanding and grasp of these skills and techniques will be reinforced throughout with practical exercises. Learners will require to test their scripts and should be given a set of test data. Scripts will be tested to ensure their correct working to meet the user's needs. Learners will amend any errors in their solutions in order to achieve robust, reliable and efficient scripts.

Software Development: Developing Small Standalone Applications — HP2N 47

This 2-SQA-Credit Unit is designed to introduce to new and emerging areas of software development. This Unit uses a modern development environment to create a series of small interactive applications that could be mobile apps, web based apps or PC based applications. As learners develop these applications, they will have the opportunity to develop software development skills including programming and user interface design.

There are four Outcomes in the Unit:

1 Describe an appropriate development lifecycle model.
2 Design a small application.
3 Construct a small application.
4 Test and deploy a small application.

Outcome 1 is more theoretical in nature and allows learners to look back at traditional lifecycle models such the Waterfall model and illustrate how such models do not really support the rapid development of small applications. Tutors could then concentrate on more appropriate iterative and incremental models such as user centred design, simpler RAD or simpler Agile models.

A possible approach would be to use a simplified version of the RAD process with three phases:

- Requirements Planning Phase
- Construction Phase
- Deployment and Testing Phase

Whatever process is selected, it is important to emphasise the incremental nature of the phases and the activities that would be iterated within the phases. Candidates should be made aware of the key development skills of analysis, design, implementation and testing.

Given that the later Outcomes do not cover the requirements planning phase, it would be sensible to explore this area in more detail, ideally with a series of short exercises that allow the candidates to develop some skills in requirements analysis.

The use of prototypes in developments could be introduced by reviewing partially constructed applications to test their usability and to try to discover additional features that users might want. This could be performed as a group activity and would also allow the candidates to learn about usability testing.

The remaining Outcomes are practical in nature and it would make sense to deliver Outcomes 2, 3 and 4 as one whereby the learners work through a series of developments gradually introducing the development skills required for designing and constructing applications. It is envisaged that Outcome 3 will prove the most challenging for the majority of learners and hence it might make sense for the majority of the developments to be based on given designs. In a similar vein testing could be introduced after the learners had successfully created a number of applications.

Suitable development environments would include App Inventor (**http://www.appinventorbeta.com/about/**). This allows learners to quickly develop Android Apps and can be used to spark their interest in software development. This environment could be used to teach and assess this Unit. Using such an environment might enable the more competent candidates to complete Unit well within the suggested time scale. This would allow you to then introduce them to more powerful android development environments such as Eclipse with the ADT Plugin (**http://developer.android.com/index.html**).

Software Development: Programming Foundations — HP2P 47

This 1-SQA-Credit Unit introduces candidates to generic fundamental programming constructs which are required as a base for software development. The Unit should also expand and consolidate the skills learned in the Unit to *Developing Software: Introduction* (HP1R 47) by reinforcing the importance of programming/scripting within computing. Developing further the importance of good design and good programming practices within programming. The learners should consolidate basic programming skills and be introduced to more complex programming program structures. The learners should demonstrate understanding of the concepts of modularity, parameter passing and objects.

There are three Outcomes in this Unit:

1 Identify different scenarios to apply programming constructs.
2 Create a number of small programs/scripts from different scenarios.
3 Demonstrate within these pieces of work good programming practice.

This Unit has been designed to allow centres flexibility in terms of the chosen development environment used for delivery. Suitable development environments could include C, C++, C# or java.

Outcome 1 is more theoretical in nature it could be used to provide learners a solid basis of why programming has evolved as it has and the current uses of programming within a number of different strands within computing.

Outcomes 2 and 3 focus on programming constructs and since. Learners do not have to provide documentation but this can be provided by the tutor for a variety of problems and the class time spent would involve the learners writing code to meet the requirements of these particular designs.

Computing: Introduction to Project Management — HP21 47

This 1-SQA-Credit Unit will enable learners to develop the basic knowledge and skills required to plan, implement, monitor, manage and report on a small scale project.

There are three Outcomes in this Unit:

1 Understand Project Management terminology.
2 Plan and implement a project plan.
3 Monitor, manage, and report change of cost/quality/time impact on a project.

In Outcome 1 will cover many of the fundamentals of project management such as the terminology of project management; the range of skills required by project managers; the stages of the project management development cycle; software available to assist the management of projects.

To aid fuller understanding of Critical path analysis it would be advisable to show/demonstrate/worked examples of how critical path analysis is carried out manually.

Discussions on project management methodologies could encompass any of the following examples (Agile, Prince2, Waterfall, ITIL, Rapid Application Development – RAD, Software Development Life Cycle – SDLC, PMBOK).

In Outcome 2 learners will learn how to plan the development and progress of a project by scheduling the phases and tasks, including resources (human and physical), milestone points, team meetings, and identifying critical and non-critical tasks. This may be achieved either manually or by making use of suitable software.

In Outcome 3 learners will be required to modify an existing project schedule in response to an external influence (cost/time/quality change). After modification you will be expected to integrate the effects of the changes into suitable documents that could facilitate effective communication to project stakeholders.

Outcomes 2 and 3: These Outcomes should be delivered in a practical fashion, ensuring that points of learning are maintained throughout the Outcomes. Learners should be encouraged to work with project specifications and use these to learn about the various aspects of project management. Outcome 3 is likely to create a significant amount of problem solving.

Mathematics for Computing 1 — HP1H 47

This is a 1-SQA-Credit Unit which teaches methods that are very useful for learners who want to be programmers. Whilst the Unit begins from first principles, in practice the contents are too much for someone to learn in the hours that will normally be allocated. Learners should have a foundation mathematics or numeracy qualification as a pre-requisite to this Unit. This Unit covers mathematical methods and skills appropriate to computing.

There are four Outcomes in this Unit:

1 Demonstrate an understanding of scientific notation and manipulate numbers in scientific notation.
2 Demonstrate an understanding of co-ordinate systems and vectors, and apply linear transformations.
3 Demonstrate a knowledge of simple functions and the ability to perform basic algebraic operations.
4 Demonstrate the application of Boolean algebra to problem situations.

In the first Outcome, learners will learn about scientific notation and rounding. This is important because it is similar to the way computers store numbers. When programming, learners need to be aware of what is happening in the computer memory when declaring variables.

The second Outcome covers Co-ordinates and transformations. These are the skills required to create computer graphics.

Outcome 3 focuses on functions and algebra. This is useful when you have to use a computer to perform calculations or manipulate numbers. In the last Outcome, you will be introduced to design with logic gates. Computers are based entirely on logic gates and seeing how a few gates can be combined to perform a useful function gives a good insight into the workings of a computer.

Team Working in Computing — HP1X 47

This 1-SQA-Credit Unit will provide the opportunity to develop effective skills for team working in the context of computing. Learners will develop co-operative working skills which will include negotiation of goals, roles and responsibilities in the development of a team based Information and Communication Technology (ICT) project. Learners, both individually and as a team, will present the project Outcomes within the timescale prescribed by the team. Individual progress will be tracked against a project plan and the team will develop skills in updating the plan to ensure that the project is delivered on time. Individual team members will contribute to any necessary research and to documentation of the group's activities.

Learners will also develop skills in evaluation and will be required to critically evaluate the contributions of themselves and fellow team members.

This Unit is aimed at developing the necessary skills for effective team working within the computing industry.

There are three Outcomes in this Unit:

1 Effectively participate in planning and organising a co-operative ICT project.
2 Participate in the management of a co-operative ICT project and research and carry out agreed project tasks against the schedule and within the remit of the project role.
3 Review own and group skills demonstrated throughout the co-operative ICT project.

Computing: Graded Unit 1 — HP2A 47

This 1-SQA-Credit Unit will provide evidence that learners have met the aims of the first year of the Group Award. It assesses the student's ability to integrate knowledge and skills and problem solving across four of the first year Units.

Year 2 Units

Software Development: Object Oriented Programming — HP2L 48

This 2-SQA-Credit Unit is designed to cover object oriented programming skills. It is a non-introductory Unit and assumes prior knowledge and proficiency in basic programming concepts and techniques. Completion of Year 1 software development Units is a recommended pre-requisite for this Unit.

In this Unit learners will acquire knowledge of the concepts, principles, and techniques of object oriented software development necessary to enable then to design and develop object oriented software.

There are three Outcomes in this Unit:

1 Investigate object oriented programming techniques and apply them to a design.
2 Implement a solution from an object oriented design using object oriented techniques.
3 Test the completed product.

This will involve the following areas of learning:

Using the features of an object oriented programming language, to implement a software solution based on a given design.

The features of OOP should include:

- Object oriented concepts and terms
- Object oriented programming techniques
- Objects and classes
- Attributes and methods
- Parameter passing
- Abstraction, encapsulation and information-hiding
- Inheritance
- Polymorphism
- Association
- Aggregation and collection
- Coupling and cohesion

Understanding and grasp of object oriented concepts and programming techniques will be reinforced throughout with practical exercises.

Using a test plan to test software to ensure it works correctly and meets the user requirements. Learners will be required to amend any errors in their code in order to achieve a robust, reliable and efficient working program.

Systems Development: Object Oriented Analysis and Design — HP2M 48

This 2-SQA-Credit Unit is designed to enable learners to develop a knowledge of the theoretical concepts, underlying principles, scope and role of systems analysis and design undertaken within an object oriented environment.

The Unit develops practical systems development skills and will introduce a variety of requirements, engineering techniques and the main modelling and diagramming techniques used in object oriented systems analysis and design, using UML or similar modelling notation.

The Unit will also develop an appreciation of the boundaries, strengths and limitations of object oriented systems analysis and design so that you can select the most appropriate tools and techniques for undertaking analysis and design given a specific project context.

There are three Outcomes in this Unit.

1 Describe the object oriented paradigm.
2 Produce a static model of a system.
3 Produce a dynamic model of a system.

This Unit has been developed with the intention of taking the learners through all phases of analysis and design from requirements gathering through to specifying the physical design of the system under consideration. Although it is envisaged that UML or a similar modelling notation is used to model and document analysis and design, the Outcomes of the Unit are generally specified to allow the Unit to be delivered using selected techniques from combinations of object oriented analysis and design methods. The choice of techniques and modelling notation may be matched to the requirements of the stakeholders, the type of system under consideration and development environment.

Outcome 1 is theoretical in nature and develops an understanding of the main concepts of UML, analysis and design within the object oriented life cycle.

Outcome 2 addresses requirements gathering and the creation of the static system model. The high level specification of the client's requirements should indicate all the users/actors that interact with each case of usage, and the relationship between cases of usage. This may be done using a diagram such as UML Use Case Diagram. These can then be modelled using Class, Responsibility, Collaboration (CRC) Cards. These cards can then be used in the creation of the class (conceptual) diagram for Outcome 3.

Outcome 3 involves creating a dynamic model of the system. This is important to show how objects will need to interact in order to perform important system processes. As with all modelling, it is important to focus on modelling that is most beneficial either for problem solving or communication, rather than steadfastly producing every possible type of diagram whether it is relevant or not. The Evidence Requirements therefore ask for use case diagrams and scenarios, and sequence diagrams.

Software Development: Data Structures — HP2K 48

Data structures are used in all but the simplest developments and hence are a topic that learners will need to understand to develop efficient software applications.

There are four Outcomes in this Unit

1 Describe data representation and storage in computer systems.
2 Describe and use data structures.
3 Describe, develop and use abstract data types.
4 Use Standard Collection classes to implement object oriented designs.

The Unit starts by exploring how different types of data are stored in computer systems and how the use of standards allows data to be transferred from one system to another system.

The Unit then introduces a range of data structures that can be used to store collections of data and illustrates how these can be used to implement some common abstract data types such as Queues and Lists. Learners will learn how the efficiency of searching for and sorting data can be improved using a range of algorithms.

The final Outcome of the Unit will allow further enhancement of programming skills by teaching use of generic collection class libraries provided in development environments. These skills are essential to program object oriented designs.

The Unit is assessed using three assessments. The first assessment is a knowledge based assessment that helps to ensure that the background knowledge required for developing skills in using data structures. The second assessment looks at ability to follow an algorithm. The final assessment consists of a number of implementation exercises that enable development of implementation skills.

Scripting for Interactivity — HP2D 48

This is a 2-SQA-Credit Unit developing multimedia scripting Modern multimedia packages offer a wide range of facilities, but there are times when the built in features do not support a particular effect or activity. This Unit introduces you to the scripting facilities provided by the multimedia packages in order to extend their range. These scripts (effectively small programs) are particularly useful when implementing systems that require interactivity.

There are four Outcomes in this Unit:

1 Select a multimedia authoring tool for a script driven application.
2 Develop a system specification and detailed design for a script driven multimedia application.
3 Use the scripting facilities of a multimedia authoring tool to implement interactivity.
4 Test the completed product.

Outcome 1 introduces scripting facilities provided by packages Learners will be required to research these in order to be able to select the most appropriate package to develop a system to meet the requirements.

Outcome 2 addresses the design issues surrounding the development of a system. The assessment for this Outcome is to construct a design document for an interactive multimedia product that requires scripting to be utilised.

Outcome 3 covers the implementation phase of a system. For this Outcome learners will be required to actually build the system, including the interface elements, and multimedia elements that are to be included and the scripts that control the behaviour of the system.

The final Outcome of this Unit covers the testing phase of product development. Learners will be required to develop a testing strategy, and then perform the tests to ensure that the system does not contain flaws.

Self Describing Data (XML) — HP2H 48

This 1-SQA-Credit Unit is designed to provide learners with the knowledge of the use of self describing data in communicating data between a wide range of applications.

The learners will learn practical skills in the generation of XML documents and the use of tools such as XML editors and XML generation tools to produce these documents. The Unit introduces the syntax of well-formed XML documents, the use of validation techniques and concludes with the transformation and styling of XML documents, suitable for publication in a desired format.

There are three Outcomes in this Unit:

1 Create well-formed XML documents.
2 Validate XML documents.
3 Transform XML documents.

This Unit is designed to teach the underpinning knowledge required for developing XML based applications in an online environment. This Unit introduces the fundamentals of XML, including how to structure a basic XML document, validate XML documents, link XML documents and transform XML documents into a variety of formats.

This Unit will introduce the role of XML documents in distributing data in a common standardised format. XML documents can be transformed into many different formats for many platforms.

This Unit will introduce transforming XML into XHTML for the publication of web pages.

The fundamental knowledge will provide skills to develop XML based technologies such as:

- RSS (Really Simple Syndication)
- SMIL(Synchronised Multimedia Integration Language)
- SVG (Scalable Vector Graphics)
- VXML (VoiceXML)

Software Development: Rapid Applications Development and Prototyping — HP2F 48

This 2-SQA-Credit Unit is designed to expose learners to the development of a software product using recognised techniques within a modern well-defined methodology. The aim of the Unit is to increase the candidates' skills in designing and building software applications using fourth generation (HG) or object oriented languages.

There are four Outcomes in this Unit:

1 Describe the components of Rapid Application Development (RAD) environments.
2 Produce a working functional/design specification.
3 Carry out design and redesign of an application.
4 Test each completed prototype.

This Unit has been developed with a view towards software development using VBA, code generators rather than towards the more formal languages such as Java. However, formal-programming languages can be adopted. The intention of the Unit is to keep the Outcomes as generic as possible to allow the Unit to be delivered using any of the fourth generation languages or application developers that are available.

Web Development: Dynamically Generated Content — HP2T 48

This is a 2-SQA-Credit Unit which is designed to teach the underpinning knowledge required for dynamic data driven web application development and gives you the opportunity to apply that knowledge when planning, designing and developing a dynamic web application with data driven content using a server side language.

There are four Outcomes in this Unit:

1 Investigate dynamic data driven web development.
2 Plan and design a dynamic data driven website.
3 Implement a dynamic data driven website.
4 Test and deploy a dynamic data driven website.

Students will learn about the types of web applications, various development languages, server configurations, databases and development environments that can be used for developing dynamic data driven web applications.

Students will learn how to plan and design a dynamic data driven website. You will learn how to develop a dynamic data driven web site that involves web applications using a server side language and a data source. The server side language may be one of PHP, ASP, ASP.NET, ColdFusion or JSP.

Students will be introduced to development through a framework or Integrated Development Environment as well as writing raw code. The IDE could be one of Visual Web Developer Studio or Dreamweaver/ColdFusion

Finally students will learn how to test and deploy a valid dynamic data driven website.

Computing: Software Development: Graded Unit 2 — HP2R 48

This 2-SQA-Credit Unit will provide evidence that learners have met the aims of the Group Award.

It will be assessed by a practical project. The project will involve planning, implementation and evaluation of a piece of software to meet user requirements. The project will involve analysis and design, implementation and testing and documenting the solution.

6.3 Opportunities for integration of Units

It is envisaged that where possible centres will deliver this award in an integrative manner to help the learners appreciate the interconnections between the various subjects.

Integration means identifying opportunity either within a Unit or across Units to deliver and/or assess topics which meet the criteria for either two Outcomes within the same Unit, or two Outcomes — one Outcome from one Unit and one Outcome from another Unit.

For example the two Units *Developing Software: Introduction* (HP1R 47) and *Software Development: Programming Foundations* (HP2P 47) both require that learners create and test software using programming constructs and creating documentation. Learners have to demonstrate good practice in their code and documentation. The practical developmental elements can be combined and delivered and assessed as an integrated Unit. The Unit *Software Development: Programming Foundations* (HP2P 47) has additional content in that learners have to demonstrate an understanding of the evolution and appropriate of software development languages and also compare object oriented and non-object oriented programming. This content will be delivered separately.

SQA Advanced Diploma

Year 1

Unit code	Unit title	Integration opportunity
HP1R 47	Developing Software: Introduction	The content of Developing Software Introduction can be wholly integrated with the Programming Foundations Unit. Teaching can be integrated completely across the two Units. For assessment the same scenario, design and test plan can be used to assess both Units.
HP2P 47	Software Development: Programming Foundations	
HP2J 48	Relational Database Management systems	In the Relational Database Unit learners are required to create a relational database using Data Definition Language (DDL) and to access and manipulate data using Data Manipulation Language (DML). It is likely that SQL would be the chosen language. Outcomes 1 and 2 of the SQL Unit require that learners create tables and query the data within the tables. Integration therefore is expected with these two Units and they should be delivered in together and also the common content assessed as an integrated assessment.
HP2E 47	SQL: Introduction	

Year 2

Unit code	Unit title	Integration opportunity
HP2L 48	Software Development: Object Oriented Programming	These two Units are designed to be delivered in tandem there would be a natural integration of content. A greater learning experience can be achieved if learners are encouraged to design the programming examples that they code. In the programming Unit learners are required to prepare a static deign model as the assessment for Outcome 1. This is also a requirement of the Analysis and Design Unit Outcome 2.
HP2M 48	Systems Development: Object Oriented Analysis and Design	

7 Assessment in an SQA Advanced Diploma award

7.1 Assessment in learning and for certification

Assessment is the process of evaluating a student's learning.

Assessment takes place throughout the learning and teaching processes as well as the final assessment for certification. It can take many forms (for example: practical exercises, case studies, extended response questions) and can be used for different purposes — including identifying prior knowledge, identifying gaps in learning, providing feedback to learners as well as measuring student attainment.

Assessment as part of the learning process is called **formative** assessment. It provides developmental feedback to a student and tutors so that they can adjust their plan for future learning. It is not recorded for external purposes. **Formative** assessment is often called 'assessment for learning'.

Summative assessment is carried out for the purpose of certification. Through **summative assessment**, learners provide evidence to demonstrate that they can achieve the Evidence Requirements detailed in the statement of standards of the relevant Unit specification. It is generally undertaken at the end of a learning activity or programme of learning and is used to make a judgement on the student's overall attainment.

7.2 Assessment planning of an SQA Advanced Diploma

All SQA Advanced Diploma Qualifications are **summatively assessed** using a mix of continuous Unit assessment and Graded Unit assessments. It is helpful for learners, the Course Team and the internal verifiers if the Course Team has an overview of when summative assessments are likely to occur. It is, therefore, common practice for a Course Team, prior to the start of course delivery to agree the overall learning, teaching and assessment plan for the course. Part of this process requires tutors to agree when each Unit in the course will be **summatively assessed**.

In situations where Units of a course are being delivered in parallel, it is important that Course Teams make sure that the assessment load placed on learners is manageable, although it is recognised that by its very nature summative assessments will occur towards the end of learning.

7.3 Planning the Unit summative assessment

For each Unit, it is helpful for tutors/assessors to draw up a Unit assessment plan which:

- describes what is to be assessed.
- says what assessment methods will be used.
- describes how the assessments are to be administered, eg practical, online, etc.
- defines opportunity for integrating assessment.
- provides a timetable for when the assessment will take place.
- notes arrangements that need to be made to take account of additional support needs or prior learning.
- describes the measures to be taken to ensure that the evidence produced is authentic and current.
- describes how and when requirements for record-keeping and quality assurance processes will be met.

7.4 Negotiating summative assessments with the learners

Ultimately, it is up to the tutor to determine when a student is ready for summative assessment (within the agreed time constraints of the course timetable). A good way of gauging if a student is ready for assessment is to use a **practice assessment** (a final formative assessment which mirrors the summative assessment in terms of assessment method and an aspect of the Evidence Requirement where appropriate but it must not contain the same task detail as the summative assessment).

The tutor can use this assessment to identify the level of an individual student's competence and the Outcome can help the tutor determine if the student is ready to attempt the summative assessment or if the student still has gaps in knowledge and understanding that need to be addressed through further work.

It is good practice to communicate assessment plans to learners as early as possible in the course so that they know what to expect. A copy of the **proposed Course Assessment Plan** may be given to learners at the start of the course, often during course induction. Thereafter, it is up to each tutor to make sure that learners receive early warning of when assessment is likely to take place.

7.5 Summative assessment exemplars

Assessment exemplars are produced by SQA and are made available to centres for Units in this SQA Advanced Diploma. Assessment exemplars are intended solely for the purpose of assessment of learners against the standards given in the Unit specifications. **They must not be released prior to the assessment or be distributed for any other purpose. It is the centre's responsibility to maintain the security of all assessment exemplars.**

A Unit assessment exemplar will contain:

- details of the conditions under which the assessment is to be carried out.
- assessment tasks for each Outcome.
- a marking scheme or model answer.
- checklists (where appropriate).

It is vital that tutors:

- adhere to the conditions for the assessment, ie open-book, closed-book, controlled conditions.
- mark assessments consistently in line with the marking scheme or model answer provided.
- keep all assessment exemplars secure so that they can be used for future student assessments.

Once the student has completed the summative assessment, it is good practice for tutors to mark their student's work quickly and provide constructive feedback.

8 SQA Advanced Diploma in Computing: Software Development assessment strategy and plan

Assessors must ensure authenticity of assessment. Generally by carrying out assessment under supervised conditions this is sufficient to ensure that candidates submit their own work.

For candidates assessment is an important part of the learning process. Candidates should be given notice of assessment so that they can be properly prepared for the assessment. Once marked candidates should be given appropriate feedback which will allow them to evaluate their own progress.

8.1 SQA Advanced Diploma in Computing: Software Development assessment strategy

A guide to the type and number of assessments in each Unit of the SQA Advanced Diploma in Computing: Software Development is shown below.

	Assessment − Year 1			
Unit	**Outcome 1**	**Outcome 2**	**Outcome 3**	**Outcome 4**
Developing Software: Introduction HP1R 47	Open-book Practical task undertaken in supervised conditions over an extended period of time			
Professionalism and Ethics in Computing HP29 47	Open-book Single assessment for Unit relating to a case study and questions covering content of Outcomes Undertaken supervised conditions over an extended period of time			
Computer Systems Fundamentals HP1T 47	Open-book Set of 25 short-response questions covering each bullet point Supervised assessment lasting 2 hours		Open-book Practical task and detailed log Supervised conditions with no time limit	
Troubleshooting Computer Problems HP1V 47	Open-book Investigation carried out and all stages presented in a 1000 word report Undertaken supervised conditions over an extended period of time			
Database Design Fundamentals HP2G 47	Open-book Practical task Create relational data model from given requirements	Open-book Practical task Use SQL to implement data model from Outcome 1	Open-book Practical task Use SQL to query database from Outcome 2	

SQA Advanced Diploma

	Assessment − Year 1			
Unit	**Outcome 1**	**Outcome 2**	**Outcome 3**	**Outcome 4**
	The three Outcomes are linked in assessment by a single requirements specification. The assessments should be carried out in supervised conditions over an extended period of time			
Relational Database Management Systems HP2J 48	Closed-book 20 multiple-choice questions Supervised conditions lasting 1 hour	Open-book Practical task covering content of Outcomes 2, 3 and 4 Can be carried out as a group task Supervised conditions over an extended period of time		

	Assessment − Year 1			
Unit	**Outcome 1**	**Outcome 2**	**Outcome 3**	**Outcome 4**
SQL: Introduction HP2E 47	Open-book Practical task Creation of tables Supervised conditions with no time limit	Open-book Practical task Creation of queries Supervised conditions with no time limit	Open-book Practical task Creation of reports Supervised conditions with no time limit	
Software Development: Developing Small Standalone Applications HP2N 47	Closed-book 20 multiple-choice questions Supervised conditions lasting 1 hour	Open-book Practical task covering content of Outcomes 2, 3 and 4 Supervised conditions over an extended period of time		
Software Development: Programming Foundations HP2P 47	Open-book Series of practical task demonstrating content of Unit Supervised conditions over an extended period of time			
Computing: Introduction to Project Management HP21 47	Closed-book 20 multiple-choice questions Supervised conditions lasting 1 hour	Open-book Practical task covering content of Outcomes 2 and 3 Supervised conditions over an extended period of time		

SQA Advanced Diploma

<table>
<tr><th></th><th colspan="4">Assessment − Year 1</th></tr>
<tr><th>Unit</th><th>Outcome 1</th><th>Outcome 2</th><th>Outcome 3</th><th>Outcome 4</th></tr>
<tr><td rowspan="4">Mathematics for Computing 1

HP1H 47</td><td>Open-book
Set of questions
Supervised conditions lasting 45 minutes</td><td>Open-book
Set of questions
Supervised conditions lasting 45 minutes</td><td>Open-book
Set of questions
Supervised conditions lasting 45 minutes</td><td>Open-book
Set of questions
Supervised conditions lasting 45 minutes</td></tr>
<tr><td colspan="4">OR</td></tr>
<tr><td colspan="4">Open-book
Set of questions for complete Unit. Learners may be given questions one week in advance
Supervised conditions with no specified time limit</td></tr>
<tr><td colspan="4">Open-book
Group project over an extended period of time</td></tr>
<tr><td>Computing: Graded Unit 1 HP2A 47</td><td colspan="4">Closed-book
Examination paper lasting 3 hours
Supervised conditions</td></tr>
</table>

SQA Advanced Diploma

<table>
<tr><th></th><th colspan="4">Assessment − Year 2</th></tr>
<tr><th>Unit</th><th>Outcome 1</th><th>Outcome 2</th><th>Outcome 3</th><th>Outcome 4</th></tr>
<tr><td>Software Development: Object Oriented Programming

HP2L 48</td><td colspan="3">Open-book
Practical project covering entire content of the Unit
Supervised conditions over an extended period of time</td><td></td></tr>
<tr><td>Systems Development: Object Oriented Analysis and Design

HP2M 48</td><td>Closed-book
20 multiple-choice questions
Supervised conditions lasting 1 hour</td><td colspan="2">Open-book
Practical project covering entire content of the Unit
Supervised conditions over an extended period of time</td><td></td></tr>
<tr><td>Software Development: Data Structures

HP2K 48</td><td>Closed-book
30 multiple-choice questions
Supervised conditions lasting 1 hour</td><td>Open-book
Practical task
3 separate desk check exercises
Supervised conditions lasting 30 minutes each</td><td>Open-book
Practical tasks
Supervised conditions over an extended period of time</td><td></td></tr>
<tr><td>Scripting for Interactivity

HP2D 48</td><td>Open-book
Research/presentation activity
Supervised conditions with no time limit</td><td colspan="3">Open-book
Practical project
Supervised conditions over an extended period of time</td></tr>
<tr><td>Self Describing Data (XML)

HP2H 48</td><td colspan="3">Open-book
Practical project covering entire content of the Unit
Supervised conditions over an extended period of time</td><td></td></tr>
<tr><td rowspan="2">Software Development: Rapid Applications Development and Prototyping

HP2F 48</td><td rowspan="2">Closed-book
20 multiple-choice questions
Supervised conditions lasting 1 hour</td><td>Open-book
Practical task
Create design from given requirements</td><td>Open-book
Practical task
Implement design Outcome 2</td><td>Open-book
Practical task
Test implementation from Outcome 3</td></tr>
<tr><td colspan="3">The three practical Outcomes are linked in assessment by a single requirements specification. The assessments should be carried out in supervised conditions over an extended period of time</td></tr>
</table>

<table>
<tr><th></th><th colspan="4">Assessment − Year 2</th></tr>
<tr><th>Unit</th><th>Outcome 1</th><th>Outcome 2</th><th>Outcome 3</th><th>Outcome 4</th></tr>
<tr><td rowspan="2">Web Development: Dynamically Generated Content

HP2T 48</td><td>Closed-book
20 multiple-choice questions
Supervised conditions lasting 1 hour</td><td colspan="3">Open-book
Practical task from given requirements</td></tr>
<tr><td></td><td colspan="3">The Outcomes are linked in assessment by a single requirements specification
The assessments should be carried out in supervised conditions over an extended period of time</td></tr>
<tr><td>Computing: Software Development: Graded Unit 2

HP2R 48</td><td colspan="4">Open-book
Practical project covering entire content of the Unit
Supervised conditions over an extended period of time</td></tr>
</table>

8.2 Course Assessment Plan

Suggested course assessment schedules for Year 1 and Year 2 are found in Appendices 2a and 2b.

They are based on two Semesters of 17 weeks, where centres have a different length of Semester, they should amend their schedules accordingly.

If centres decide to deliver Units in a different order, they should amend their schedules accordingly.

Much of the assessment in the SQA Advanced Diploma in Computing Software Development is based on practical skills. The assessment plan indicates the time when the assessment should be complete and submitted. It is likely that candidates will be working on the assessment for number weeks prior to submission. Evidence for practical assessment may be gathered as a portfolio of evidence. A portfolio will allow candidates to build up a body of evidence by submitting small sections as they are completed. This method of assessment allows the candidate to submit evidence and receive feedback on an ongoing basis. This can be useful for candidates as they can use the feedback in the next stage of the assessment.

Dates indicated on the assessment timing grid represent the expected completion date of all assessment for the Unit. Time is allowed for resubmission.

8.3 SQA Advanced Diploma in Computing: Software Development Graded Unit assessments

Computing: Graded Unit 1 (HP2A 47) Examination at SCQF level 7 — 1 SQA Credit

Computing: Software Development Graded Unit 2 (HP2R 48) Practical Project at SCQF level 8 — 2 SQA Credits

Computing: Graded Unit 1 is a closed-book examination lasting three hours comprising three sections. The examination assesses the candidate's critical knowledge and understanding of the topics relating to the specific aims which this Graded Unit is designed to cover. The questions and corresponding marks are designed in accordance with the ranges indicated in the table that follows. However, the overall total mark for the examination is 100.

The question paper consists of three sections, totalling 100 marks (100%).

Section 1 will be worth 15% of the total marks.
Section 2 will be worth 15% of the total marks.
Section 3 will be worth 70% of the total marks.

The sections will be differentiated by content and level of demand which will be reflected in the type of question used in each section.

Section	Type of question	No of questions	Marks per question	Total marks
1	Selected response	15	1	15
2	Constructed response	1 (mandatory)	15	15
3	Constructed response	7 from 8	10	70

Section 1 — Selected response questions		
Key topics	**Level of demand**	**Percentage weighting for each topic**
Computer Systems Fundamentals	Ability to demonstrate fundamental knowledge and understanding introduced in the three Units.	Five questions each worth 1 mark (5% of total)
Developing Software: Introduction		Five questions each worth 1 mark (5% of total)
Professionalism and Ethics in Computing		Five questions each worth 1 mark (5% of total)

Section 2 — Constructed response questions		
Key topics	**Level of demand**	**Percentage weighting for each topic**
Integrated question incorporating at least two of the following Units ♦ Computer Systems Fundamentals ♦ Developing Software: ♦ Introduction ♦ Professionalism and Ethics in Computing ♦ Troubleshooting Computing Problems	Application, analysis, synthesis and evaluation	One question worth 15 marks (15% of total)

Section 3 — Constructed response questions		
Key topics	**Level of demand**	**Percentage weighting for each topic**
Computer Systems Fundamentals	Knowledge, comprehension, application and analysis	Two questions each worth 10 marks.
Developing Software: Introduction	Knowledge, comprehension, application, analysis, synthesis and evaluation	Two questions each worth 10 marks.
Professionalism and Ethics in Computing	Knowledge, comprehension, application, analysis, synthesis and evaluation	Two questions each worth 10 marks.
Troubleshooting Computing Problems	Knowledge, comprehension, application and analysis	Two questions each worth 10 marks

Note: The candidate will choose seven from eight questions in section 3, 70% of the total.

This assessment should take place towards the end of the programme to ensure that learners have covered the topics which will be assessed within the Graded Unit.

Graded Unit 2 is a project investigation to be completed on an open-book basis over a period of time.

SQA Advanced Diploma

This Unit covers the integration of a range of knowledge and skills achieved throughout selected Units of the SQA Advanced Diploma:

Software Development: Object Oriented Programming (HP2L 48)
Systems Development: Object Oriented Analysis and Design (HP2M 48)
Software Development: Data Structures (HP2K 48)

The project brief should include a sample of topics and issues selected from the following list of Outcomes from mandatory Units. The assessor may want to consider some suggestions in the table below. (To be completed when Units are finalised)

Unit code	Unit title	Topics/Issues
HP2L 48	Software Development: Object Oriented programming	♦ Investigate object oriented programming techniques and apply them to a design. ♦ Implement a solution from an object oriented design using object oriented techniques. ♦ Test the completed product.
HP2M 48	Systems Development: Object Oriented Analysis and Design	♦ Describe the object oriented paradigm. ♦ Produce a static model of a system. ♦ Produce a dynamic model of a system.
HP2K 48	Software Development: Data Structures	♦ Describe data representation and storage in computer systems ♦ Describe and use data structures ♦ Describe, develop and use abstract data types ♦ Use Standard Collection classes to implement object oriented designs

Note: The list of Topics/Issues in the above table is not exhaustive. Depending on the characteristics of the project brief, the assessor may draw Outcomes from other Units in the SQA Advanced framework provided such Units were undertaken by the candidate.

In addition to the integration of knowledge and skills needed to complete the project investigation, learners will develop their skills in planning, negotiation, research, analysis, time management and problem solving. The Core Skill of *Problem Solving* at SCQF level 6 is embedded and therefore automatically certificated on successful completion of the Unit.

8.4 What happens if a learner does not achieve an assessment?

If a student fails to demonstrate competence in a summative assessment, it is good practice to communicate this to the student quickly. Tutors should take time to individually feedback to learners where they went wrong. Having given feedback, tutors should then advise learners on what they need to do to prepare for re-assessment.

The student then undertakes additional work as discussed with the tutor, this is called remediation. It is when the student revises class work or practices skills covered in class BEFORE they attempt the re-assessment. It is important that learners do get time to consolidate their knowledge and understanding before being re-assessed.

Re-assessment may take a variety of forms.

- For some assessments, learners may be allowed to provide additional information, eg if a student has submitted a report based on a piece of independent research, s/he would be allowed to add the missing evidence and resubmit the report. The new information should be highlighted in such a way to show that it had been added, eg underlined, coloured and dated in the margin.
- For practical tasks related to their use of information technology, learners may be permitted to correct work and resubmit — the original submission and the resubmission should both be kept.
- For multiple-choice, short-response and Graded Units, learners may be required to attempt a completely new assessment instrument.

Where specific action has to be taken for re-assessment details will be noted in the Unit specification and tutors must be familiar with the Unit specification requirements for re-assessment.

It is important to note that re-assessment does **NOT** always require that learners complete a full new assessment. Re-assessment may (and often does) allow learners to re-attempt the part of the Outcome that they have not completed to a standard which meets the Unit specification.

For Units other than Graded Units, SQA provides only **ONE** summative assessment and it is extremely important that centres produce their own **alternativ**e assessments. These assessments can be used for re-assessment purposes.

Once a draft assessment has been prepared by the centre it should first be quality checked by centre staff (internally verified) and submitted to SQA for prior-verification to ensure that it is fit for purpose.[3]

If a student fails to reach the pass mark in the Graded Unit 1 — examination then he/she should be allowed to sit an alternative examination before the beginning of the next session to allow progression to Year 2. Learners must complete all aspects of the new assessment instrument.

[3] For centres wishing support in this process, an online course has been developed − Produce HN Unit assessments for successful prior moderation

9 Quality Assurance

SQA is committed to providing qualifications and support to match the needs of individuals, society and the economy of Scotland and internationally. SQA believes that global interaction in education and training benefits our customers, clients, partners and SQA through the sharing of expertise and experience.

SQA has a balanced portfolio of qualifications that is inclusive, facilitates progression, reflects Scotland's educational, economic, social and cultural needs and changes, and supports education and training worldwide.

SQA works in partnership with our approved centres to achieve our shared goals of excellence and consistency. This ensures that SQA's qualifications continue to meet the requirements all users of our qualifications.

SQA's quality assurance models are designed to ensure that assessment decisions made to national standards are correct and consistent, and that national standards are maintained. We are committed to maintaining an assessment and quality system that is easy to understand, effectively administered, publicly accountable, and cost-effective to operate.

As well as working with centres to manage and enhance the quality of SQA qualifications, SQA routinely monitors its own performance. SQA establishes processes that need to be followed and submit these to regular auditing. This includes systematic evaluation and review of the effectiveness of our quality management processes. SQA also monitors standards across all our qualifications over time, to ensure consistency. Feedback from stakeholders is an integral part of SQA's review activities.

SQA is subject to external audit by a number of agencies, including the Scottish Government.

For assessed qualifications, SQA regularly monitors centres to ensure that they have the resources and expertise to assess learners against the qualification criteria. (Assessment is where centres use assessment instruments to make decisions about learners' work. External assessment is where SQA takes on these duties, usually in the form of examinations or externally assessed coursework. Only assessments are subject to quality assurance by verification.)

10 Verification

10.1 Introduction

SQA's quality assurance processes have been developed to ensure that national standards are applied to internally assessed Units or course components.

To maintain the credibility of SQA qualifications, we rely on effective collaboration with centres to ensure national standards are maintained across all qualifications at all levels.

Verification is the procedure that SQA uses to make sure that centres' assessment decisions are valid and reliable and are in line with national standards.

10.2 Why do we need verification?

Verification is one of a range of Quality Assurance measures used by SQA to confirm that:

- centres' assessment decisions are sound (ie valid, reliable and practicable).
- national standards are being uniformly applied.
- assessments are accurately and consistently applied across all learners and levels.

This ensures qualifications and certification is credible with all learners being assessed to a common standard.

10.3 Internal verification

Centres are responsible for the internal verification of their assessments. This means that centres should have an internal verification system — a system of having quality checks in place — which can be operated throughout the centre. Each tutor who is responsible for the assessment of learners and/or internal verification of student material should:

- be made aware of their centre's quality assurance procedures.
- comply with these procedures.

Centres will appoint staff members to be internal verifiers. Internal verifiers will ensure that assessors apply standards of assessment uniformly and consistently. They should keep records of internal verification activity for external verifiers to access. Examples of records include:

- evidence of planned verification for the Semester which conforms to the centre's verification strategy.
- minutes of meetings where assessment work is examined and where discussion about acceptable standards is noted and decisions recorded.
- internal verification forms showing which learners' work has been verified and the Outcome. Note where an assessor carries out observations, internal verifiers should also observe the assessor.
- evidence of discussion and support of assessors, particularly where student work has not been accepted by the internal verifier.
- evidence of reporting back to the Course Team, any recommendations/actions required and evidence that these are acted upon.

10.4 External verification

To ensure national consistency in assessment decisions, SQA appoints experienced teachers/lecturers who have good, recent experience in the delivery and assessment of their subject to carry out external verification in centres. SQA will notify the SQA Co-ordinator if your centre has been selected for verification.

SQA wants to encourage centres and staff to see verification in a positive light, as a valuable Quality Improvement tool.

Appendix 1a: Core Skills Year 1

Unit code	Unit title	Communication	Numeracy	ICT	Problem Solving	Working with Others
		Written Communication Oral Communication	**Using Number Using Graphical Information**	**Accessing Information Providing/Creating Information**	**Critical Thinking Planning and Organising Reviewing and Evaluating**	**Working Co-operatively with Others Reviewing Co-operative Contribution**
HP1R 47	Developing Software: Introduction				Critical thinking Embedded (SCQF level 6)	
HP29 47	Professionalism and Ethics in Computing	Signposted (SCQF level 6)				
HP1T 47	Computer Systems Fundamentals		Embedded (SCQF level 5)			
HP1V 47	Troubleshooting Computer Problems				Embedded (SCQF level 6)	
HP2G 47	Database Design Fundamentals				Signposted (SCQF level 5)	
HP2J 48	Relational Database Management Systems			Signposted (SCQF level 6)	Signposted (SCQF level 6)	
HP2E 47	SQL: Introduction					
HP2N 47	Software Development: Developing Small Standalone Applications				Signposted (SCQF level 6)	

SQA Advanced Diploma

Unit code	Unit title	Communication	Numeracy	ICT	Problem Solving	Working with Others
		Written Communication Oral Communication	**Using Number Using Graphical Information**	**Accessing Information Providing/Creating Information**	**Critical Thinking Planning and Organising Reviewing and Evaluating**	**Working Co-operatively with Others Reviewing Co-operative Contribution**
HP2P 47	Software Development: Programming Foundations			Signposted (SCQF level 6)	Signposted (SCQF level 6)	
HP21 47	Computing: Introduction to Project Management					
HP1H 47	Mathematics for Computing 1		Embedded Using Number (SCQF level 6) Using Graphical Information (SCQF level 5)			
HP1X 47	Team Working in Computing	Signposted (SCQF level 6)		Embedded (SCQF level 6)		Embedded (SCQF level 6)
HP2A 47	Computing: Graded Unit 1					

Appendix 1b: Core Skills Year 2

Unit code	Unit title	Communication	Numeracy	ICT	Problem Solving	Working with Others
		Written Communication Oral Communication	**Using Number Using Graphical Information**	**Accessing Information Providing/Creating Information**	**Critical Thinking Planning and Organising Reviewing and Evaluating**	**Working Co-operatively with Others Reviewing Co-operative Contribution**
HP2L 48	Software Development: Object Oriented Programming				Signposted (SCQF level 6)	
HP2M 48	Systems Development: Object Oriented Analysis and Design				Signposted (SCQF level 6)	
HP2K 48	Software Development: Data Structures				Signposted (SCQF level 6)	
HP2D 48	Scripting for Interactivity					
HP2H 48	Self Describing Data (XML)	Written communication Signposted (SCQF level 6)			Signposted (SCQF level 6)	
HP2F 48	Software Development: Rapid Applications Development and Prototyping					

SQA Advanced Diploma

Unit code	Unit title	Communication	Numeracy	ICT	Problem Solving	Working with Others
		Written Communication Oral Communication	**Using Number Using Graphical Information**	**Accessing Information Providing/ Creating Information**	**Critical Thinking Planning and Organising Reviewing and Evaluating**	**Working Co-operatively with Others Reviewing Co-operative Contribution**
HP2T 48	Web Development: Dynamically Generated Content	Signposted (SCQF level 5)			Signposted (SCQF level 5)	
HP2R 48	Computing: Software Development: Graded Unit 2				Embedded (SCQF level 6)	

Appendix 2a: Year 1, Semester 1 — Assessment Plan (Option A)

Unit name\Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Developing Software: Introduction (HP1R 47)																X*	
Professionalism and Ethics in Computing (HP29 47)														X*			
Computer Systems Fundamentals (HP1T 47)										X					X		
Database Design Fundamentals (HP2G 47)								X				X				X	
Software Development: Programming Foundations (HP2P 47)									X		X					X*	
Computing: Introduction to Project Management (HP21 47)													X*		X		
Troubleshooting Computer Problems (HP1V 47)															X*		

Submission date is shown by an X, X* indicates integrated assessment and Outcomes integrated.
Shaded cells indicates closed-book assessment.

Appendix 2a: Year 1, Semester 2 — Assessment Plan (Option A)

Unit name\Week	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34
Team Working in Computing (HP1X 47)															X*		
Computing: Graded Unit 1 (HP2A 47)												X					
Relational Database Management Systems (HP2J 48)						X*				X*				X*	X		
Software Development: Developing Small Standalone Applications (HP2N 47)												X*		X			
SQL: Introduction (HP2E 47)						X*				X*				X*			
Mathematics for Computing 1 (HP1H 47)				X				X				X				X	

Submission date is shown by an X, X* indicates integrated assessment and Outcomes integrated.
Shaded cells indicates closed-book assessment.

Appendix 2b: Year 2, Semester 1 — Assessment Plan (Option A)

Unit name\Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Software Development: Object Oriented Programming (HP2L 48)																X*	
Systems Development: Object Oriented Analysis and Design (HP2M 48)															X	X*	
Software Development: Data Structures (HP2K 48)								X						X	X		
Software Development: Rapid Applications Development and Prototyping (HP2F 48)					X				X				X	X			

Submission date is shown by an X, X* indicates integrated assessment and Outcomes integrated.
Shaded cells indicates closed-book assessment.

Appendix 2b: Year 2, Semester 2 — Assessment Plan (Option A)

Unit name\Week	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34
Scripting for Interactivity (HP2D 48)					X									X*			
Self Describing Data (XML) (HP2H 48)														X*			
Web Development: Dynamically Generated Content (HP2T 48)															X		
Computing: Software Development: Graded Unit 2(HP2R 48)			X								X				X		

Submission date is shown by an X, X* indicates integrated assessment and Outcomes integrated.
Shaded cells indicates closed-book assessment.