SQA Advanced Unit Specification

General information for centres

Unit title: High Level Engineering Software

Unit code: HP41 47

Unit purpose: This Unit is designed to give candidates knowledge and understanding and apply basic software engineering concepts to solving electrical and electronic engineering problems that require a software solution. The emphasis in this Unit is on writing and testing and documenting I/O programs using the basic structures available in most high level languages.

This Unit provides the candidates with the opportunity to develop skills in system design (top-down design), detailed design (using flow charts or program design language), implementation in the design in a high level language and verification of the design. This Unit also allows the candidates to develop their communication skills by generating the documentation for the designed solution.

On completion of the Unit the candidate should be able to:

1 Write, test and document linear programs using I/O statements.
2 Write, test and document I/O programs using conditional statements.
3 Write, test and document I/O programs using iterative loop statements.
4 Write, test and document an I/O program, incorporating subroutines.

Credit value: 1 SQA Credit at SCQF level 7: (8 SCQF credit points at SCQF level 7*)

**SCQF credit points are used to allocate credit to qualifications in the Scottish Credit and Qualifications Framework (SCQF). Each qualification in the Framework is allocated a number of SCQF credit points at an SCQF level. There are 12 SCQF levels, ranging from National 1 to Doctorates.*

Recommended prior knowledge and skills: Candidates should have a basic knowledge of electronics, logic operations and number systems. This may be evidenced by possession of a Higher in Electronics or the following National Qualification Units D134 11 *Combinational Logic* or E9LG 11 *Computing in Engineering 1* or D980 11 *Programmable Systems.*

Core Skills: There may be opportunities to gather evidence towards Core Skills in this Unit, although there is no automatic certification of Core Skills or Core Skills components.

Context for delivery: This Unit was developed for the SQA Advanced Certificate/Diploma in Electronics award. If the Unit is used in another Group Award(s) it is recommended that it should be taught and assessed within the context of the particular Group Award(s) to which it contributes. The platform for this Unit may be a micro-controller system or a standard personal computer with access to I/O ports.

Assessment: It is recommended that this Unit be assessed via a logbook or portfolio and programming projects or assignments that cover the knowledge and skill presented in each Outcome.

Centres should take every reasonable effort to ensure that reports are the candidates' own work. It may be possible to issue each candidate with a slightly different specification of equal complexity, or alternatively where there is a suspicion of copying or plagiarism, an appropriate response may be to interview candidates. A checklist should be used to record oral evidence of the candidates' understanding.

Centres are recommended to develop appropriate checklists to support the assessment requirements for each of the knowledge and skills items. Centres are also recommended to produce a marking schedule based on the Evidence Requirements listed indicating clearly the required content of the report. Candidates who do not meet the standard should be obliged to correct and resubmit their work.

Unit specification: statement of standards

Unit title: High Level Engineering Software

Unit code: HP41 47

The sections of the Unit stating the Outcomes, knowledge and/or skills, and Evidence Requirements are mandatory.

Where evidence for Outcomes is assessed on a sample basis, the whole of the content listed in the Knowledge and/or Skills section must be taught and available for assessment. Candidates should not know in advance the items on which they will be assessed and different items should be sampled on each assessment occasion.

Outcome 1

Write, test and document linear programs using I/O statements

Knowledge and/or Skills

- Software engineering process
- Creating and editing files (word processing)
- Constants and variables
- Data types: integer, real, character, string, Boolean and array
- Operators: +, -, *, and /
- Statements: input, output
- Documentation

Evidence Requirements

Evidence for the Knowledge and/or Skills in this Outcome will be provided on a sample basis in the form of a portfolio. The candidate’s response will be judged to be satisfactory where evidence provided is sufficient to show that the candidate is able to:

Write, test and document ONE linear program involving:

- input and output statements
- different types of variables
- arithmetic operators

Outcome 2

Write, test and document I/O programs using conditional statements

Knowledge and/or Skills

- Relational operators: ==, !=, >=, <=, <, >
- Logical operators
- Boolean expressions
- Branch statements: IF…THEN, IF…THEN…ELSE, CASE…OF
- Ports: configuring, reading and writing

Evidence Requirements

Evidence for the Knowledge and/or Skills in this Outcome will be provided on a sample basis in the form of a portfolio. The candidate’s response will be judged to be satisfactory where evidence provided is sufficient to show that the candidate is able to:

Write, test and document ONE non-linear I/O program involving:

- conditional statements with logic operators
- multi-way selection statement

Outcome 3

Write, test and document I/O programs using iterative loop statements

Knowledge and/or Skills

- Flowcharts/pseudo code: FOR...DO, REPEAT…UNTIL, WHILE… loops
- Statements: FOR...DO, REPEAT…UNTIL, WHILE…
- Loops to manipulate data arrays:
  - output array data to a port
  - read port data into an array
- Testing of iterative loops
- Test plan and actual results

Evidence Requirements

Evidence for the Knowledge and/or Skills in this Outcome will be provided on a sample basis in the form of a portfolio. The candidate’s response will be judged to be satisfactory where evidence provided is sufficient to show that the candidate is able to:

Write, test and document ONE non-linear program involving an iterative loop

- the program must include a conditional statement inside the iterative loop OR and iterative loop must be a part of the conditional statement

Outcome 4

Write, test and document an I/O program, incorporating subroutines

Knowledge and/or Skills

- Procedures/functions
- Bit-wise logical operations: AND, OR and EOR
- Identifying port signals and their functions
- Test plan and actual results
- Documentation

Evidence Requirements

Evidence for the Knowledge and/or Skills in this Outcome will be provided on a sample basis in the form of a portfolio. The candidate’s response will be judged to be satisfactory where evidence provided is sufficient to show that the candidate is able to:

Write, test and document a basic input/output ports program, which includes:

- procedures/functions
- bit-wise logical operation.
- a conditional statement
- an iterative loop

the documentation to include:

- statement of the problem
- flowchart/algorithm
- program coding with comments
- testing and actual results
- conclusion

Administrative information

Unit code:	HP41 47
Unit title:	High Level Engineering Software
Superclass category:	CB
Original date of publication:	August 2017
Version:	01

History of changes:

Version	Description of change	Date

Source: SQA

© Scottish Qualifications Authority 2004, 2009, 2017

This publication may be reproduced in whole or in part for educational purposes provided that no profit is derived from reproduction and that, if reproduced in part, the source is acknowledged.

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

FURTHER INFORMATION: Call SQA’s Customer Contact Centre on 44 (0) 141 500 5030 or 0345 279 1000. Alternatively, complete our Centre Feedback Form.

Unit specification: support notes

Unit title: High Level Engineering Software

This part of the Unit specification is offered as guidance. The support notes are not mandatory.

While the exact time allocated to this Unit is at the discretion of the centre, the notional design length is 40 hours.

Guidance on the content and context for this Unit

This Unit has been written in order to allow candidates to develop the knowledge, understanding and skills in the area of Software Engineering by writing and testing I/O programs. The following processes illustrate Software Engineering:

- user requirements
- problem specification
- system design (partitioning into manageable tasks)
- detailed design (design of individual modules)
- implementation (coding into chosen programming language)
- program verification (software testing)
- program maintenance (extending the program for the future)

With validation and documentation taking place at all the stages.

This Unit was designed to permit the candidates to apply the software engineering techniques to formulate the solutions to engineering tasks involving input and output ports. Software may be developed for a PC or single board computer or a micro-controller in conjunction with an applications board. Candidates may be provided with subprograms to configure and read/write input/output ports.

This Unit was developed as one of four SQA Advanced Certificate/Diploma in Engineering Programming options and is at SCQF level 7. This Unit should be completed before tackling the level 8 Unit, *High Level Language: External I/O Transfer*.

The Unit has identified the topics that are expected to be covered by lecturers. The Unit also gives recommendations as to how much time should be spent on each Outcome. This has been done to help lecturers to decide what depth of treatment should be given to the topics attached to each of the Outcomes. Whilst it is not mandatory for a centre to use this list of topics it is strongly recommended that they do so to ensure continuity of teaching and learning across the Engineering Computing Units and because the assessment exemplar pack for this Unit is based on the Knowledge and/or Skills and list of topics in each of the Outcomes.

A list of topics is given below. Lecturers are advised to study this list in conjunction with the assessment exemplar pack for this Unit so they can get a clear indication of the standard of achievement expected of candidates taking the Unit.

1 **Write,** test and document linear programs using I/O statements **(10 hours)**

User Requirements and analysis
Problem Specification
Top-down design
Hardware

Inputs (ports, switches, analogue to digital converter)

Outputs (ports, motors, LEDS, digital to analogue converters)

Platform (personal computer, development system, micro-controller)

Software

System Design (partitioning into manageable tasks)

Structure Chart

Input, process, output

Modular programming and step-wise refinement

Detailed design using pseudo code or **flow charts**

Linear programming using ports

Creating and saving source file

Compiling, and debugging source code syntax errors

Executing the compiled code (run)

2 **Write**, test and document I/O programs using conditional statements **(8 hours)**

Selection

```
IF condition THEN
        actions if condition holds
IF condition THEN
        actions if condition holds
ELSE
        actions if the condition does not hold
END_IF
CASE condition OF
        case 1: action(s) for case 1
        case 2: action(s) for case 2
.
.
        case n: action(s) for case n
END_CASE
```

Conditions based on input PORT values.

Use of built in procedures such as **delay, sound, clrscr**

Compilation to DOS platform

3 **Write,** test and document I/O programs using iterative loop statements**.** **(10 Hours)**

Repetition

```
FOR variable {FROM start} TO finish {IN STEPS OF value}
DO
        actions to be repeatedly performed
END_FOR
REPEAT
        actions to be performed
UNTIL a condition holds


WHILE condition to be tested DO
        actions to be performed
END_WHILE
```

Preparation of test data to determine the principal software function
Preparation of test data to test all decision paths
Preparation of (unusual) data to determine the robustness of the program
Testing of program (single step, symbolic debugger and output statements)
Recording of test results
Analysis of the results and corrective action

Candidates should be encouraged to fully test and document the test results of selected tutorial exercise of Outcome 3 in a tabular format.

4 **Write,** test and document an I/O program, incorporating subroutines **(12 hours)**

Programs with input/output ports

- Ports and control register addresses
- Configuring ports
- Reading input port
- Writing output port
- Masking

Program with procedures/functions

- Global variables (no parameter passing)
- Execution of procedures/functions

This Outcome should be delivered through the use of sample programs. The execution of these should be demonstrated to the candidates; in particular the input/output ports programs.

The practical tutorials should allow students to modify existing programs rather than starting from scratch. The tutorial exercises should be developed in the context of the Outcome 4 assignment.

The need and importance of documentation
Use of unusual/special instructions or procedures
Changes in personnel
Program maintenance
Operator instructions
Hardware requirements
Documentation, a continuous process in the software development life-cycle
Standard format for documentation

Guidance on the delivery and assessment of this Unit

This is the first engineering programming Unit on the SQA Advanced Certificate/Diploma Electronics and for many candidates this may be their first encounter with programming. In this respect it may be advisable to deliver this Unit on a free-standing basis.

Each candidate should have access to a computer loaded with appropriate OS and High Level Language software. Additionally workstations should be available on which candidates are able to test I/O based programs.

Delivery of all the Outcomes may be carried out using an LCD projector so that the candidates can view the actual code of the sample programs and see their execution with prepared test data in real terms.

This introductory Unit should be used to develop the candidate’s confidence in the basic programming concepts and as such should be taken before considering assembly language programming.

It is suggested that the Unit assessment take the form of a logbook or portfolio covering the four Outcomes.

For example, assuming a two-hour class per week, then over a 20-week period the assessment schedule may be:

Weeks 1 to 5	Linear programming Outcome 1 assignments completed **(2hours)**
Weeks 6 to 10	Non-linear programming for I/O ports using selection statements Outcome 2 assignments completed **(2hours)**
Weeks 11 to 15	Non-linear I/O programming using iterative loops Outcome 3 assignments completed **(2hours)**
Weeks 16 to 20	I/O programming using bit-wise operations Outcome 4 assignments completed **(4hours)**

Open learning

This Unit is more suitable for laboratory delivery however it could be delivered by distance learning provided the candidate has the appropriate hardware. This may require some degree of on-line support. However, with regard to assessment, planning would be required by the centre concerned to ensure the sufficiency and authenticity of candidate evidence. Arrangements would be required to be put in place to ensure that assessment whether done at a single or at multiple events was conducted under controlled, supervised conditions.

To keep administrative arrangements to a minimum, it is recommended that a single assessment paper (taken by candidates at a single assessment event) be used for distance learning candidates.

Equality and inclusion

This unit specification 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.

General information for candidates

Unit title: High Level Engineering Software

This is the first of four Engineering Programming Units in the SQA Advanced Certificate/Diploma Electronics. This Unit introduces you to I/O programming in the context of the Software Engineering Process. This process allows you to break down the overall programming task into manageable sub tasks.

These smaller modules are then developed, debugged and tested using a High Level Language. The overall program is then constructed by integrating the modules one by one and testing as the program builds up. The final program is tested for functionality and each branch is exercised using appropriate test data.

This Unit teaches you:

- system design (partitioning of a software task)
- detailed design (stepwise refinement using pseudo code or flowcharts)
- coding in a High Level Language
- compiling and debugging
- preparing test data and verifying program function
- documentation

In addition you will learn and practise the following programming constructs:

- linear programming
- IF…THEN, IF…THEN…ELSE, CASE…
- FOR…NEXT…DO
- REPEAT…UNTIL
- WHILE…DO
- procedure/functions
- logical bit-wise operations

You will be presented with a variety of programs, but the emphasis will be on driving or controlling hardware interface circuitry. Hence you will also cover INPUT/OUTPUT ports, configuring them, writing to them and reading from them.

The time allocated for this Unit is 40 hours. If you are new to programming then you will probably require to spend double this amount as the practical activities can be time consuming. It is in your interest not just to pass the Unit for qualification needs but to master it, as most electronics related jobs and higher study demand the skills and knowledge embedded in this Unit.

Your assessment for this Unit will be assignments covering each of the four Outcomes, to be carried out in the context of the Software Engineering Process. Class time allocated for this will be about 14 hours, however you will be expected to devote the same number of hours in private study for the assignments. The written assignments must conform to the standard set out by the lecturer. Outcome 4 must be submitted within 14 days of completion of the Unit.