Higher National Unit Specification

General information

Unit title: Programming for Data (SCQF level 8)

Unit code: J4YB 35

Superclass:	CA
Publication date:	December 2020
Source:	Scottish Qualifications Authority
Version:	02

Unit purpose

This unit is designed to enable learners to become familiar with the concepts, principles and practices around data programming. Learners should have previous experience of data analysis. It would also be desirable for learners to possess programming skills and an understanding of basic statistics.

This is a **non-specialist** unit, intended for a wide range of learners; it is particularly appropriate for learners with an interest in computing and its associated sub-disciplines, including data science.

This unit covers: Integrated Development Environments, programming languages for data analysis, functional programming, project set-up, reproducibility and good programming practice, command line and data interfaces, version control, data security, data types and structures, automation, data manipulation, statistical functions, performing data analysis tests and using APIs.

On completion of this unit, learners may progress to J27J 35 *Computer Programming* at SCQF level 8, for a more thorough grounding in coding skills, or J4YC 36 *Data Engineering* at SCQF level 9.

Outcomes

On successful completion of the unit the learner will be able to:

1 Describe the processes and technologies used in a data programming environment.
2 Explain the concepts, processes and practices used to create clean, efficient code for data analysis.
3 Write functions and programs to perform data analysis.
4 Test programs for data analysis.

Higher National Unit Specification: General information (cont)

Unit title: Programming for Data (SCQF level 8)

Credit points and level

1 Higher National Unit credit at Scottish Credit and Qualifications Framework (SCQF) level 8: (8 SCQF credit points at SCQF level 8)

Recommended entry to the unit

While entry is at the discretion of the centre, the learner should have experience of data analysis concepts and be familiar with both programming concepts and statistics. In understanding data analysis, units such as J4Y4 34 *Working with Data* (level 7) would be beneficial. It would also be advantageous to have completed J0HA 34 *Computer Programming* at SCQF level 7.

Core Skills

Achievement of this Unit gives automatic certification of the following:

Complete Core Skill	Problem Solving at SCQF level 6

There are also opportunities to develop aspects of Core Skills which are highlighted in the support notes of this unit specification.

Context for delivery

If this unit is delivered as part of a group award, it is recommended that it should be taught and assessed within the subject area of the group award to which it contributes.

This unit could be delivered after J4Y6 35 *Working with Data* at SCQF level 8. The knowledge and skills gained by taking that unit would provide good under-pinning for this unit.

This unit focuses on data programming, rather than more general programming (or software development) skills. Units such as J0HA 34 *Computer Programming* (SCQF level 7) and J27J 35 *Computer Programming* (SCQF level 8) provide better introductions to general programming.

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**.

Higher National Unit Specification: Statement of standards

Unit title: Programming for Data (SCQF level 8)

Acceptable performance in this unit will be the satisfactory achievement of the standards set out in this part of the unit specification. All sections of the statement of standards are mandatory and cannot be altered without reference to SQA.

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. Learners 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

Describe the processes and technologies used in a data programming environment.

Knowledge and/or skills

- Software types for data analysis
- High level programming languages for data analysis including functional languages
- Database languages including SQL
- Low code software development tools including notebooks
- Integrated Development Environments
- Command Line Interfaces
- Application Programming Interfaces
- Software libraries
- Programming process
- Source/version control

Outcome 2

Explain the concepts, processes and practices used to create clean, efficient code for data analysis.

Knowledge and/or skills

- Setting up a project including software installation
- Programming techniques
- Data sources
- Data types and data structures
- Library functions for data analysis
- Data interfaces and data flow
- Data modelling
- Data security/secret management
- Code reuse
- Algorithmic bias and data bias

Unit title: Programming for Data (SCQF level 8)

Outcome 3

Write functions and programs to perform data analysis.

Knowledge and/or skills

- Writing function/program to process data from an external file
- Writing function/program to process data from a database
- Writing function/program to clean data
- Writing function/program to wrangle data
- Writing function/program to perform statistical analysis
- Writing function/program to create visualisations
- Writing documentation for function/program

Outcome 4

Test programs for data analysis.

Knowledge and/or skills

- Test plans
- Testing datasets
- Training datasets
- Debugging
- Code review
- Performing test and revising code

Unit title: Programming for Data (SCQF level 8)

Evidence requirements for this unit

Learners will need to provide evidence to demonstrate the knowledge and/or skills across all outcomes. The evidence requirements for this unit will take **one** form.

- Product evidence

The **product evidence** will relate to all outcomes and will take the form of **at least two** programs for data analysis. The evidence must demonstrate that the learner can:

- set up the development environment using the appropriate processes and technologies
- select appropriate data structures and data models
- utilise the concepts, processes and practices needed to create quality, reusable code
- test and debug the code
- write documentation

The programs must include:

1. reading data from external sources (at least two)
2. data cleaning algorithms
3. data wrangling
4. statistical analyses
5. visualisations

The code must be error free and produce correct results.

This evidence may be produced over the life of the unit, under loosely controlled conditions (including access to reference materials). When evidence is produced in loosely controlled conditions it must be authenticated. The *Guide to assessment* provides further advice on methods of authentication.

The SCQF level of this unit (level 8) provides additional context on the nature of the required evidence and the associated standards. Appropriate level descriptors should be used when making judgements about the evidence.

The support notes section of this specification provides specific examples of instruments of assessment that will generate the required evidence.

Higher National Unit Support Notes

Unit title: Programming for Data (SCQF level 8)

Unit support notes are offered as guidance and 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 focuses on the importance of writing structured clean logical code following good programming practices. The unit is suitable for a wide range of learners with an interest in the practicalities of using code for data analysis.

In outcome 1 the focus will be on exploring the programming languages used to create the programs for data analysis. Learners will look at the range of languages used in data analysis: Python, R, Matlab, Scala, etc. Although the learner should be exposed initially to a range of languages, it is essential that they then focus on just one programming language.

The learner will look at where datasets are stored (text files, spreadsheets, etc) and how they are read into the programs utilising the programs read, functions and libraries. For example, Python’s CSV module. They will also explore data types; how data is stored in the language; and how it flows through the program. For example, in Python, using Pandas to replace an Excel™ table with data frames. Emphasis should be placed on the use of modularity, using functions, libraries and code reuse.

In outcome 2 the focus is on the importance of setting up a proper data programming environment. The learner will look at options for setting up, installing and configuring different development environments. For example, to utilise Python programming, the set-up could include setting up Anaconda Python distribution as a package and environment manager, and an IDE like Atom™ or Jupyter™ Notebooks. If using R programming language, then the learner may use R Studio as an IDE and GIT for source control. Learners will also utilise command line, such as Unix Shell to automate tasks. For example, using grep, awk and sed commands along with git.

Alongside the advantages of good programming practice (standardised naming conventions, code documentation, etc), the learner should also discover the importance of securing data, including the tools and functions used for managing digital authentication (secret management).

Unit title: Programming for Data (SCQF level 8)

Outcome 3 is practical in nature and involves the learner writing a series of functions and programs. The learner should follow the processes and structures they learned in outcomes 1 and 2 when writing their programs. The functions/programs they write should include:

- reading data from external sources (at least two)
- data cleaning algorithms
- data wrangling
- statistical analyses
- visualisations

Outcome 4 is also practical in nature and involves the learner setting up tests for their programs, writing the code to run the test and reviewing the test after completion.

Guidance on approaches to delivery of this unit

It is anticipated that learners will complete the outcomes sequentially. A suggested distribution of time, across the outcomes, is:

Outcome 1:	10 hours
Outcome 2:	10 hours
Outcome 3:	14 hours
Outcome 4:	6 hours

In outcome 1 the learner should be exposed to a range of development environments, languages and types of interface. There are opportunities for presentations, classroom discussions as well as group work. There are opportunities for the learner to consolidate their learning with practical exercises, eg setting up IDEs, interacting with version control software (Git, Bit Bucket), working with command line and installing appropriate software. There are also opportunities to research advantages and disadvantages of each language and approaches to programming with data.

In outcomes 2, 3 and 4 the learner should focus on one programming language and one development environment. It is not a prerequisite for the learner to necessarily have programming experience. Therefore, time should be taken over the key concepts needed to create efficient code. The learner should be familiar with these concepts to assist in the writing of their programs/functions.

Guidance on approaches to assessment of this unit

Evidence can be generated using different types of assessment. The following are suggestions only. There may be other methods that would be more suitable to learners.

Centres are reminded that prior verification of centre-devised assessments would help to ensure that the national standard is being met. Where learners experience a range of assessment methods, this helps them to develop different skills that should be transferable to work or further and higher education.

Unit title: Programming for Data (SCQF level 8)

There are a number of opportunities to carry out formative assessment through every stage in the unit. Assessors will then have an opportunity to intervene if the learner has misconceptions or lacks clarity on particular topics. Diagnostic feedback can be provided to learners on an ongoing basis.

It is recommended that all the outcomes within the unit are assessed using an e-portfolio. E-portfolios give learners the opportunity to take ownership of their own work and for assembling the e-portfolios themselves. This working e-portfolio could continuously be updated by the learner as they gather different materials. However, it is important that learners understand that, in order to achieve the unit, the e-portfolio they submit for assessment should contain the items that clearly recognise and record their achievement.

Examples of evidence that could be used in the e-portfolio:

- Screenshots of installing software, interacting with GIT, setting up IDEs
- Blog posts: Reflecting on programming languages, different interfaces, etc
- Presentations: (audio/video)
- Code listings, documentation, test plans, test results, created charts, interaction with source control software, etc

The programs must include:

1 reading data from external sources (at least two)
2 data cleaning algorithms
3 data wrangling
4 statistical analyses
5 visualisations

E-portfolios allow different types of electronic evidence to be used for assessment in its original format. They offer an assessment approach that is inherently 'learner-centred' and offer more flexibility in that learners can assemble one portfolio and tailor it to specific audiences by tagging items for different purposes (including different assessments). They allow evidence to be stored in a manner that is more secure and more accessible to learners, assessors and verifiers. They make it easier for assessors to give feedback, which strengthens the links between formative and summative assessment, and between learning and assessment generally.

Opportunities for e-assessment

E-assessment may be appropriate for some assessments in this unit. By e-assessment we mean assessment which is supported by Information and Communication Technology (ICT), such as e-testing or the use of e-portfolios or social software. Centres which wish to use e-assessment must ensure that the national standard is applied to all learner evidence and that conditions of assessment as specified in the evidence requirements are met, regardless of the mode of gathering evidence. The most up-to-date guidance on the use of e-assessment to support SQA's qualifications is available at **www.sqa.org.uk/e-assessment**.

Unit title: Programming for Data (SCQF level 8)

Opportunities for developing Core and other essential skills

This unit will provide opportunities for learners to develop Core Skills in *Problem Solving* and *Information and Communication Technology (ICT)* at SCQF level 6. All three components of *Problem Solving* (Critical Thinking, Planning and Organising, Reviewing and Evaluating) could be developed in the unit.

The Core Skill of Problem Solving at SCQF level 6 is embedded in this unit. When a learner achieves the unit, their Core Skills profile will also be updated to include this Core Skill.

History of changes to unit

Version	Description of change	Date
02	Core Skill Problem Solving at SCQF level 6 embedded.	09/12/20

Additional copies of this unit specification can be purchased from the Scottish Qualifications Authority. Please contact the Business Development and Customer Support team, telephone 0303 333 0330.

Unit template: June 2017

General information for learners

Unit title: Programming for Data (SCQF level 8)

This section will help you decide whether this is the unit for you by explaining what the unit is about, what you should know or be able to do before you start, what you will need to do during the unit and opportunities for further learning and employment.

This unit is designed to help you become familiar with the concepts, principles and purposes around programming for data. The primary focus is on writing good quality, reusable, clean code needed to interrogate data.

It is particularly appropriate if you have an interest in computing and its associated sub-disciplines, including data science.

Some experience of performing data analysis is required for this unit, and it would be advantageous also to have a familiarity with programming concepts and an understanding of basic statistics.

This unit covers the following topics, among others:

- setting up of a data analysis environment
- understanding data flow
- functional programming
- data types and structures
- programming concepts, structures and processes used to write quality, reusable code
- designing a test plan for a program
- testing and debugging code
- writing documentation

You will have opportunities to write complex programs in a designated language common in data analysis, eg Python, R and MatLab.

Assessment will likely be through a range of assessment methods, all of which will be practical in nature.

On completion of this unit, you may progress to J4YD 36 *Machine Learning* at SCQF level 9 or J4YC 36 *Data Engineering* at SCQF level 9.

The Core Skill of Problem Solving SCQF at level 6 is embedded in this unit. When a learner achieves the unit, their Core Skills profile will also be updated to include this Core Skill.