

Next Generation Higher National Unit Specification

Event-driven Programming (SCQF level 8)

Unit code: J7EC 48
SCQF level: 8 (16 SCQF credit points)
Valid from: session 2023–24

Prototype unit specification for use in pilot delivery only (version 1.0) June 2023

This unit specification provides detailed information about the unit to ensure consistent and transparent assessment year on year.

This unit specification is for teachers and lecturers and contains all the mandatory information required to deliver and assess the unit.

The information in this unit specification may be reproduced in support of SQA qualifications only on a non-commercial basis. If it is reproduced, SQA must be clearly acknowledged as the source. If it is to be reproduced for any other purpose, written permission must be obtained from permissions@sqa.org.uk.

This edition: June 2023 (version 1.0)

© Scottish Qualifications Authority 2023

Unit purpose

This is a specialist unit, intended for learners with an interest in computer programming and the development of software applications (apps) that require event-handling, such as graphical user interfaces (GUIs) and web-based front-ends.

The unit covers a range of competencies, including:

- ◆ designing the user front-end for a software app
- ◆ designing, coding and testing the appropriate components to handle user interactions
- ◆ using an event-driven programming platform to:
 - create and test code
 - create accessible user interfaces
 - conduct user experience (UX) surveys
 - evaluate findings to improve interface usability

Before starting the unit, learners should have a basic understanding of computer programming at SCQF level 7, including a knowledge of variables, programming constructs and the software development life cycle.

On completion of the unit, learners can design a GUI or web interface and create program code to enable it to function to satisfy user requirements.

Unit outcomes

Learners who complete this unit can:

- 1 design a front-end event-driven app to meet user requirements
- 2 implement a front-end event-driven app
- 3 perform and report testing of an event-driven app
- 4 evaluate the findings from UX testing
- 5 deploy a front-end event-driven app

Evidence requirements

Learners must provide product evidence. Knowledge is inferred from the product evidence.

The product evidence must demonstrate that learners can:

- ◆ design a software solution for an event-driven front-end app
- ◆ implement the solution in an appropriate event-driven framework
- ◆ develop a test plan, test the solution, and document test results
- ◆ prepare technical and user documentation for the solution
- ◆ carry out useability testing and evaluate results
- ◆ deploy the software

You should assess learners individually. They must create at least one complete, working, front-end app that meets the evidence requirements. The problem should be non-trivial and sufficiently complex to satisfy the SCQF level. The problem may be simulated or based on a real-world task or client-based problem. You should use the same problem for all stages of assessment.

Learners must conform to the software development life cycle by producing:

- ◆ a requirements specification
- ◆ design documentation
- ◆ evidence of implementation and testing

You should emphasise accessibility and UX. Learners must conduct a survey of users and carry out an evaluation of feedback, and use this to identify improvements to the UX.

The code must be sufficiently complex to demonstrate understanding of the key features of:

- ◆ event-driven systems
- ◆ events and event handlers
- ◆ control structures

NextGen: HN published prototype unit specification for use in pilot delivery only (version 1.0)
June 2023

- ◆ modular programming
- ◆ parameter passing
- ◆ input and output

Learners should demonstrate good practice in programming throughout the unit. They must produce both technical and user documentation for the solution and an evaluation of the UX.

Learners can produce evidence over an extended period in lightly-controlled conditions. Evidence produced in lightly-controlled conditions must be authenticated. The [Guide to Assessment](#) provides further advice on methods of authentication.

Knowledge and skills

The following table shows the knowledge and skills covered by the unit outcomes:

Knowledge	Skills
<p>Learners should understand:</p> <ul style="list-style-type: none">◆ the software development life cycle◆ how to interpret user requirements◆ key features of event-driven systems◆ the specification and design of event-driven applications◆ tools and techniques of an event-driven language◆ state machines◆ the event-driven programming language environment, syntax and frameworks◆ event handlers◆ design patterns for event handlers◆ program control structures◆ good programming practice◆ types of events◆ UX design and accessibility	<p>Learners can:</p> <ul style="list-style-type: none">◆ identify user requirements◆ create an event-driven design◆ create a graphical event-driven interface◆ use an event-driven programming language environment◆ develop an interface◆ develop modular code◆ use event handlers◆ use exception handlers◆ create a test plan◆ carry out testing◆ document test results◆ perform UX testing◆ evaluate UX feedback◆ deploy a developed system

Meta-skills

Throughout this unit, learners develop meta-skills to enhance their employability in the software development sector.

Self-management

This meta-skill includes:

- ◆ focusing: sorting and maintaining documentation throughout development in a logical and efficient manner; attention to detail to ensure error-free, robust code; considering all aspects of user requirements
- ◆ adapting: critically reflecting on own skills development and evaluation of user experience; self-learning to develop wider skills and extend development beyond requirements and taught content
- ◆ initiative: independent thinking to establish user requirements and design solutions; developing a front-end application based on client information; self-motivation and taking ownership of own development and the need to satisfy user requirements; time management

Social intelligence

This meta-skill includes:

- ◆ communicating: receiving information to establish the user requirements and ensure an understanding of the brief; giving information during UX testing to confirm objectives and ensure that instructions are understood
- ◆ feeling: storytelling through the creation of technical documentation, providing a walk-through and detail of the system

Innovation

This meta-skill includes:

- ◆ creativity: using imagination to provide a solution that meets the needs of both the application and the user; seeing the UX through the eyes of the user; idea generation and thinking about problem areas and how to provide a solution; visualising to create an overall impression of the completed solution throughout the process
- ◆ sense-making: analysis; seeing the bigger picture
- ◆ critical thinking: logical thinking to ensure a coherent approach and to meet requirements appropriately

Delivery of unit

The timing of delivery and assessment of this unit is at your centre's discretion.

Based on 80 hours delivery and assessment time, we suggest the following distribution:

Outcome 1 — Design a front-end event-driven app to meet user requirements
(15 hours)

Outcome 2 — Implement a front-end event-driven app
(35 hours)

Outcome 3 — Perform and report testing of an event-driven app
(10 hours)

Outcome 4 — Evaluate the findings from UX testing
(10 hours)

Outcome 5 — Deploy a front-end event-driven app
(10 hours)

Professional recognition

Although this unit is not directed towards any specific vendor exam, the materials presented are essential for further study in user interface programming. You should consider the unit a valuable pre-requisite for more advanced study.

Additional guidance

The guidance in this section is not mandatory.

Content and context for this unit

Learners require access to resources, including:

- ◆ an event-driven programming language with a supporting development platform
- ◆ a framework of tools relating to the chosen language
- ◆ a software tool that supports the design, development and debugging of code

You should give teaching time to developing event-driven apps to familiarise learners with the event-driven language and development platform. You should focus on developing learners' skills in reading code and using debugging tools to ensure that the code is robust and error free. You should focus coding activities on UX so that learners understand the importance of an accessible front-end that meets the needs of the end user. You should introduce this at an early stage.

You should include learning activities in a logical structure to ensure a sound understanding of life cycle models and documentation. You should emphasise understanding and preparing documentation.

Approaches to assessment

We recommend that learners work in groups or teams for learning activities, but assessment is carried out individually. Learners must carry out UX testing with a group of individuals, which may be other learners or a client group.

You should assess the unit using a problem with complexity suited to SCQF level 8. You should provide a range of project briefs that represent real world situations. You may also approve a project that learners have identified themselves. Where the latter is the case, you must confirm that it has sufficient complexity. The assessment is holistic, using the same project for all stages of the software development.

There may be opportunities to deliver and assess the unit in conjunction with other units in the award.

Equality and inclusion

This unit is designed to be as fair and as accessible as possible with no unnecessary barriers to learning or assessment.

You should take into account the needs of individual learners when planning learning experiences, selecting assessment methods or considering alternative evidence.

Guidance on assessment arrangements for disabled learners and/or those with additional support needs is available on the assessment arrangements web page:

www.sqa.org.uk/assessmentarrangements.

Information for learners

Event-driven Programming (SCQF level 8)

This information explains:

- ◆ what the unit is about
- ◆ what you should know or be able to do before you start
- ◆ what you need to do during the unit
- ◆ opportunities for further learning and employment

Unit information

This unit provides you with an opportunity to develop programming skills in relation to creating applications (apps) with a user interface. This is a specialist unit for learners with an interest in computer programming and the development of software apps that require event-handling, such as graphical user interfaces (GUIs) and web-based front-ends.

The unit covers a range of skills that include designing the user front-end for a software app, and designing, coding and testing the appropriate components to handle user interactions. You also gain practical skills in using an event-driven programming platform to create and test code and implement a framework that provides tools for the creation of accessible user interfaces. In addition, you also conduct UX surveys and evaluate these to improve interface usability.

Before starting the unit, you should have some knowledge of programming at SCQF level 7, including a knowledge of variables, programming constructs and the software development life cycle. The unit develops these concepts further and introduces an appropriate event-driven framework. You become familiar with the syntax and tools of the development platform.

We expect you to create small programs to develop your skills and understanding of the concepts. You apply these to larger apps. There may be an opportunity for you to work in small groups to facilitate peer learning, but we assess you individually.

We assess your performance on your creation of at least one event-driven application that satisfies the requirements of a user specified problem. You create a design, implement it in code, and test and carry out user evaluation of the application.

On completion of the unit, you may progress to further study of software development at SCQF level 8 and higher.

Throughout the unit, you develop meta-skills covering self-management, social intelligence and innovation.

The materials presented in the unit are essential for further study in user interface programming. You should consider the unit a valuable prerequisite for more advanced study in web and software development.

Administrative information

Published: June 2023 (version 1.0)

Superclass: CB

History of changes

Version	Description of change	Date

Note: please check [SQA's website](#) to ensure you are using the most up-to-date version of this document.