Next Generation Higher National Educator Guide

Higher National Diploma Software Development

Qualification code: GV22 48

This qualification is available in a restricted delivery model from academic session 2025

This guide provides information about the Higher National Diploma (HND) to ensure consistent and transparent assessment year on year. It is for lecturers and assessors, and contains all the mandatory information you need to deliver and assess the HND.

You must read it alongside the Grading Pack.

Published: October 2025 (version 1.0)

Contents

Purpose of the qualification

Higher National Diploma (HND) Software Development addresses a specialist area in computing for which there is increasing demand, as businesses and organisations increasingly adopt digital solutions. Software development has applications across a range of disciplines, such as healthcare, finance, commerce, and entertainment. Software development is constantly evolving, with new technologies and techniques that extend the range of digital solutions to improve business efficiency.

Learners experience modern approaches to the design, development and testing of software applications using a range of programming languages and tools. They use structured methodologies to discover and analyse user requirements and design software solutions that meet those requirements.

Learners work collaboratively to develop software applications, using version control and other tools to manage the development process. They evaluate software development methodologies and choose appropriate approaches for project work, including code quality, testing, and documentation.

The qualification prepares learners for a career in this computing specialism. It develops the skills and knowledge they need to succeed in the workplace, while providing them with opportunities to develop their professional skills and apply them to diverse challenges. It also prepares learners for further study in this specialism, or in other aspects of computing, including study at degree level.

Structure

Higher National Diplomas (HNDs) are at SCQF level 8 and are made up of 120 SCQF credit points (15 SQA credits). HNDs must incorporate at least 80 credit points (10 SQA credits) at SCQF level 8.

HND Software Development incorporates a mandatory project unit (Professional Practice in Software Development at SCQF level 8) with an SQA credit value of 4.

Framework

The HND is made up of mandatory and optional units. Learners must complete all the mandatory units and 8 SQA credits (64 SCQF credit points) from the optional units.

Mandatory units

Unit code	Unit title	SQA credits	SCQF credit points	SCQF level
J7D9 48	Software Development	3	24	8
J7EE 48	Professional Practice in Software Development	4	32	8

Optional units

Unit code	Unit title	SQA credits	SCQF credit points	SCQF level
J7E1 48	Application Development for Web	3	24	8
J7EB 48	Code Security	2	16	8
J7DV 48	Database Design and Development	3	24	8
J897 48	DevOps Principles and Practice	2	16	8
J691 47	Emerging Technologies and Experiences	1	8	7
J7EC 48	Event Driven Programming	2	16	8
J7ED 48	Human Computer Interface	2	16	8
J7EG 48	Programming for Data	2	16	8
J7EF 48	Software Engineering Methods	3	24	8

There are no restricted options in this qualification.

Aims of the qualification

General aims

1. Develop a range of academic competencies at SCQF level 8, including study and research skills.
2. Develop vocational skills to prepare learners for employment in current and future roles.
3. Develop capacity to apply knowledge and skills to solve real-world problems.
4. Enable progression through the Scottish Credit and Qualification Framework (SCQF), including progression to degree-level study.
5. Develop meta-skills that complement technical and professional knowledge and skills.
6. Develop Learning for Sustainability skills, knowledge, understanding and values.

Specific aims

1. Develop understanding of the processes, paradigms and concepts that underpin the development of software solutions, including object orientation.
2. Develop a range of specialist skills in software development, including programming and databases.
3. Develop competence in applying professional skills, techniques and practices associated with contemporary software development.
4. Develop understanding of current trends in software development, such as secure development and operations, and how they are applied for the benefit of society and the economy.
5. Provide practical experience and professional behaviours in applying theory to a range of real-world contexts.
6. Prepare learners for employment in the design, development, testing and implementation of software solutions to problems.
7. Develop computational thinking, maker mentality, pattern recognition, deconstruction, logical thinking, teamworking and collaboration, and synthesis and analysis skills.

8. Gain understanding of the ethical, social and legal issues associated with software development, including those relating to diversity, inclusion and sustainability.
9. Prepare for progression to further studies in software development, or related disciplines, at SCQF level 9.

Who is this qualification for?

This qualification is suitable for learners who want to develop their knowledge and skills in the design and development of software solutions in a range of contexts such as business, engineering, administration, and health and social care. It prepares them for a career in software development by providing them with opportunities to develop their professional skills and apply them to diverse challenges. It also prepares them for further study in this or other aspects of computing, including study at degree level.

Entry to this qualification is at your centre's discretion. However, we recommend that learners have one or more of the following:

- HNC Computing (GT6G 47)
- HNC Computing (GF3E 15)

Recognising prior learning

SQA recognises that learners gain knowledge and skills through formal, non-formal and informal learning contexts. Formal learning is learning certificated by a recognised awarding or professional body. Non-formal learning includes learning such as employers' in-house training courses. Informal learning is learning based on experience from a variety of environments that is not formally assessed.

It is unlikely that a learner would have the appropriate prior learning and experience to meet all the requirements of a full HND.

You can find more information and guidance about the recognition of prior learning on SQA's website.

Articulation and progression

Learners who complete this qualification could go on to:

- other qualifications in computing or related areas
- degree-level study in software development
- further study, employment and/or training

HND Software Development is not designed to articulate to any university degree programme. It does, however, provide a sound basis for progression to a range of degree programmes or other qualifications at SCQF level 9 in software development or other topics in computer science. Learners can also progress to employment in software development junior roles.

Credit transfer arrangements

Centres can make decisions about transferring credit. They can transfer credit if the subject-related content of the units is broadly equivalent. Centres should consider the currency of a learner's achievement before transferring credit.

Recommended Core Skills entry profile

Learners should have the following Core Skills at the stated SCQF levels before starting this qualification. This information can help identify learners who may need additional support.

Core Skill	Recommended SCQF entry profile
Communication	level 6
Numeracy	level 6
Information and Communication Technology (ICT)	level 6
Problem Solving	level 6
Working with Others	level 6

How the qualification meets employer and higher education institution needs

This qualification is designed in collaboration with employers, higher education institutions (HEIs), practitioners and professional bodies to meet the sector need.

The following tables show how the qualification can benefit employers and HEIs by equipping learners with the necessary skill set:

- Table 1 shows how units map to the aims of the qualification.
- Table 2 shows how the units map to National Occupational Standards (NOS).
- Table 3 shows the assessment strategy for the qualification.

Table 1: mapping qualification aims to units

General aims

Key: aim is directly relevant to unit (X), aim is optional in this unit (O), aim is not applicable to this unit (N/A)

Unit code	Unit title	Aim 1	Aim 2	Aim 3	Aim 4	Aim 5	Aim 6
J7D9 48	Software Development	X	X	X	X	X	O
J7EE 48	Professional Practice in Software Development	X	X	X	X	X	X
J7E1 48	Application Development for Web	N/A	X	X	X	X	X
J7EB 48	Code Security	N/A	X	X	X	N/A	N/A
J7DV 48	Database Design and Development	N/A	X	X	X	N/A	O
J897 48	DevOps Principles and Practice	X	X	X	X	X	O
J691 47	Emerging Technologies and Experiences	X	X	N/A	X	N/A	X
J7EC 48	Event Driven Programming	N/A	X	X	X	N/A	N/A

Specific aims

Key: aim is directly relevant to unit (X), aim is optional in this unit (O), aim is not applicable to this unit (N/A)

Unit code	Unit title	Aim 1	Aim 2	Aim 3	Aim 4	Aim 5	Aim 6	Aim 7	Aim 8	Aim 9
J7D9 48	Software Development	X	X	X	X	X	X	N/A	X	X
J7EE 48	Professional Practice in Software Development	X	X	X	X	X	X	X	X	X
J7E1 48	Application Development for Web	N/A	X	X	X	X	X	X	X	X
J7EB 48	Code Security	X	X	X	X	X	X	X	X	X
J7DV 48	Database Design and Development	X	X	X	X	X	X	X	N/A	X
J897 48	DevOps Principles and Practice	N/A	N/A	N/A	X	N/A	N/A	N/A	X	X
J691 47	Emerging Technologies and Experiences	X	N/A	X	X	X	X	X	X	X
J7EC 48	Event Driven Programming	X	X	X	N/A	X	X	X	N/A	X

Table 2: mapping National Occupational Standards (NOS)

Unit code	Unit title	NOS codes
J7D9 48	Software Development	1.1, 2.6, 4.1, 4.0, 4.4, 4.5, 4.7, 5.1, 5.2, 5.5, 5.7
J7EE 48	Professional Practice in Software Development	1.1, 2.6, 4.1, 4.2, 4.3, 4.4, 4.5, 4.6, 4.7, 5.1, 5.2, 5.3, 5.5, 5.7, 10.2, 10.3, 10.6
J7E1 48	Application Development for Web	1.1, 5.1, 5.2, 5.7, 6.4, 8.1, 10.2, 10.3, 10.6
J7EB 48	Code Security	4.7, 5.1, 5.2, 5.3, 6.3, 6.4, 8.1
J7DV 48	Database Design and Development	1.1, 2.6, 4.1, 4.2, 4.5, 5.1, 5.2, 5.3, 8.1, 10.6
J897 48	DevOps Principles and Practice	1.1, 2.6, 4.1, 4.2, 4.3, 4.4, 4.5, 4.6, 5.3, 5.7, 10.6
J691 47	Emerging Technologies and Experiences	1.1, 10.6
J7EC 48	Event Driven Programming	4.4, 4.5, 4.6, 5.1, 5.2, 5.3, 6.4, 10.6
J7ED 48	Human Computer Interface	1.1, 2.6, 4.3, 4.6, 4.7, 5.1, 5.2, 5.5, 5.7
J7EG 48	Programming for Data	4.2, 4.5, 5.1, 5.2, 5.5, 5.7, 6.4, 10.6
J7EF 48	Software Engineering Methods	1.1, 2.6, 4.1, 4.2, 4.3, 4.4, 4.5, 4.6, 5.3, 5.7, 10.6

Table 3: assessment strategy for the qualification

Unit code	Unit title	Assessment method
J7D9 48	Software Development	• Product evidence: A project report on the conduct of a software development project that is based on a real-life problem. The project must be an individual project, and learners must provide evidence for every part of the software development process in their evaluation report.
J7EE 48	Professional Practice in Software Development	• Product evidence (team): This is an extended team project of sufficient size and complexity to require a team solution to a software development brief. The project team produces a report of how it planned and managed the project, analysed the requirements; designed and implemented a software solution and its conclusions and recommendations. • Product evidence (individual): A reflective report on the individual contribution to the project. • Performance evidence: A team presentation of the conduct of the project, the outcomes and solution. • Meta-skills: A reflective report showing that the individual has self-assessed their meta-skills, created a plan for development and carried out activities that improved and developed their meta-skills. • Learning for sustainability: A report on how a software development process or product could be made more sustainable to meet the aims of at least two selected UN SDGs.

Unit code	Unit title	Assessment method
J7E1 48	Application Development for Web	• Product evidence: Learners design and build a secure full-stack interactive web app to a given user requirement and design brief. They deploy the web app to the cloud and implement it using current front-end and back-end technologies. They demonstrate the use of version control and code repositing.
J7EB 48	Code Security	• Product evidence: Learners produce a report of a software development project that focuses on security-based use cases. The project brief must represent a real-life problem that requires a security-based design and secure coding. Learners must provide: ○ user requirements with a focus on security-based use cases ○ a threat model that includes the analysis of each of the security-based cases identified in the user requirements ○ a software application that wholly depends on the security-based cases ○ identification of threat mitigation and application of threat mitigation to the codebase ○ a test plan and results to show that the learner has effectively implemented threat mitigation

Unit code	Unit title	Assessment method
J7DV 48	Database Design and Development	• Product evidence: The product evidence consists of a: ○ normalised relational database, appropriately populated, from a given specification ○ set of queries that learners use to access, analyse and manage the data ○ NoSQL database, appropriately populated ○ set of queries that learners use to manage and analyse the NoSQL data
J897 48	DevOps Principles and Practice	• Knowledge evidence: In any form of media, learners demonstrate knowledge relating to each outcome. • Product evidence: The product evidence consists of: ○ screenshots ○ a code repository folder ○ operational logs ○ code files
J691 47	Emerging Technologies and Experiences	• Knowledge evidence: In any form of media, learners demonstrate knowledge relating to each outcome.

Unit code	Unit title	Assessment method
J7EC 48	Event Driven Programming	• The product evidence must demonstrate that learners can: ○ design a software solution for an event-driven front-end app and implement it in an event-driven framework ○ develop a test plan, test the solution, and document test results ○ carry out useability testing and evaluate results ○ deploy the software along with technical and user documentation
J7ED 48	Human Computer Interface	• Product evidence: A report on the analysis, design, prototyping and usability testing of user interfaces, to meet the requirements of a given brief.
J7EG 48	Programming for Data	• Product evidence: at least two programs that perform automated data analysis. Knowledge is inferred from the product evidence. The evidence must demonstrate that learners can: ○ set up a development environment using appropriate processes and technologies ○ select appropriate data structures and data models ○ create quality, reusable code in line with good practice ○ test and debug program code ○ produce technical and user documentation for each data analysis program

Unit code	Unit title	Assessment method
J7EF48	Software Engineering Methods	- Knowledge evidence: A report or a presentation. A sampling frame is provided to guide the use of a question paper. - Product evidence: Learners demonstrate that they can successfully apply software engineering methods and tools to a range of real-world problems. They must evidence all the skills listed in the ‘Knowledge and skills’ section of the unit specification, covering the gamut of methods and processes from establishing software requirements to the design and validation of a software solution.

Meta-skills

Every NextGen: HN Qualification gives learners the opportunity to develop meta-skills.

Meta-skills are transferable behaviours and abilities that help people to adapt and succeed in life, study and work. There are three categories of meta-skills: self-management, social intelligence and innovation. Each of these is made up of four meta-skills and a number of sub-skills.

- Self-management — focusing, integrity, adapting, initiative
- Social intelligence — communicating, feeling, collaborating, leading
- Innovation — curiosity, creativity, sense-making, critical thinking

From early in the qualification, we want learners to identify and understand the meta-skills they can develop, and to appreciate the personal and professional value of these skills. We want to support learners to continue to articulate, use and build on them long after they have achieved their qualification. In this way, we help learners to develop broad skills profiles, enabling them to thrive in a changing world.

Every NextGen: HN unit signposts opportunities for learners to develop meta-skills, and there is an assessed outcome in one of the mandatory units. When you make your whole-qualification grade decisions, you consider learners' commitment to engaging with meta-skills development.

You do not assess learners on their competence or progress in individual meta-skills. Instead, you assess them on evidence that they have engaged with a personal process of development. Meta-skills development is founded on a clear process of self-assessment, goal setting, action planning and reflective practice.

You can find meta-skills teaching, learning and assessment resources on SQA's meta-skills web page.

Meta-skills in HND Software Development

You can introduce meta-skills to learners as tools they can use in response to real-world challenges and opportunities. At SCQF level 8, you should use terminology from the Skills 4.0 model, but it is important that you develop a shared understanding with learners about meta-skills and what they mean to them, both individually and in the context of coursework, projects and sectors.

You should embed meta-skills in learning and project tasks as a context for planning, practice, and reflection. You should encourage learners to be self-aware, set active goals and monitor their progress.

The process of developing meta-skills is not linear and you should make learners active participants in their learning. At the start of the process, you should introduce meta-skills to learners and explore the concept of self-assessment with them. You should set goals and make development and evaluation plans together. The process should become cyclical, with reflective practice informing new self-awareness, goal setting and review.

Many traditional learning and teaching activities used to develop industry or sector-specific skills, knowledge and understanding also support the development of meta-skills. You can map these in course materials and resources and during learning.

Meta-skills are central to successfully engaging with and completing assignments and projects. You should encourage learners to plan how they will use and develop meta-skills in their coursework and to reflect on their success and future goals.

The role of the coach, mentor or facilitator is key to help learners understand, develop and reflect on their own meta-skills and those central to course activities, assessment projects and their target industry or sector. You and any employer partners or guest speakers could guide learners by taking on a coaching and mentoring role.

In this role, you should introduce learners to the fundamentals of reflective practice. You could use several models of reflective practice. You do not need to use a theoretical perspective. Any reference to these models should support learners' understanding of the nature and value of reflective practice in self-understanding and making change.

Introducing reflective practice can support your learners' personal development and goal setting. Frequent formative peer-to-peer, assessor, client (if appropriate) and group reflection activity can support learners through reflective practice.

Learners can focus on any meta-skills appropriate to them and their context. However, learning and teaching should also facilitate individual development. Learners have individual strengths and areas for development, and they do not have to reach a particular level in relation to meta-skills. Coursework and projects provide the context for development appropriate to the SCQF level. In these contexts, the process of development is important. You should create a clear learning plan with each learner to provide evidence of their development.

You can create descriptions of abilities and skills that relate to meta-skills with your learners. These can come from self-profiling, exploring the industry and sector, and discussion with peers and employers. You should consider the meta-skills needed to complete coursework and meet personal goals to set a context for reflection.

Exploring learning and working styles, personality traits and preferences, personal profiling and self-assessment tools can help learners to develop an understanding of their strengths and areas for development.

You can use case studies and scenario-based activities to demonstrate the value of meta-skills and how to apply them. You can provide opportunities for peer reflection. A group of learners could share experiences and reflections about how to apply meta-skills in the context of their coursework. You could adopt the role of facilitator to draw learners' attention to situations where meta-skills were or could have been applied.

Reflective discussions can focus on how and where meta-skills are being developed. Your discussions with learners could include positive recognition and guidance on future development based on previous performance. As learners progress, you could introduce industry content that requires skills like problem recognition and problem solving, both of which combine multiple meta-skills.

You can deliver the knowledge and skills for practical aspects of projects in sequence. However, learners benefit from learning and teaching that integrates meta-skills with project planning and development. This approach supports learners to engage in reflective practice throughout the project and develops their self-awareness and an appreciation for continuous learning. It also maximises your opportunities to support, coach and mentor learners through their projects.

Learning for Sustainability

Context

The United Nations (UN) 2030 Agenda for Sustainable Development, adopted by the UK in 2015, has shaped the development of Scottish, national and international sustainability policy. It sets out the UN Sustainable Development Goals (SDGs), which are central to the Scottish Government's National Performance Framework. Learning for Sustainability (LfS) is a commitment to embedding the SDGs in Scottish education.

LfS embraces global citizenship, sustainable development, social justice, human rights, climate change, biodiversity loss, equality and inclusion. Learners develop their capacity to deal with the unpredictable social, economic and environmental challenges facing our rapidly changing world.

LfS combines:

- education for sustainable development (ESD)
- global citizenship
- outdoor learning

ESD is the internationally used term for sustainability education. Although LfS has a broader remit, the terms are largely interchangeable. Colleges and universities tend to use ESD, while schools usually use LfS. Both focus on a broad range of social, economic and environmental themes and approaches across all levels of education. SQA uses LfS as an umbrella term.

Learning for Sustainability in Next Generation Higher National Qualifications

Sustainability is a core component in this qualification.

Learners who complete this qualification should have:

- a general understanding of social, economic and environmental sustainability
- a general understanding of the SDGs
- a deeper understanding of subject-specific sustainability
- the confidence to apply the skills, knowledge, understanding and values they develop in the next stage of their life

Sustainability is embedded as an outcome in the Professional Practice in Software Development.

Learners who complete this outcome can:

- assess their own knowledge and understanding of sustainability and the SDGs
- review unit content against the SDGs to identify a sustainability-related issue
- apply knowledge and understanding of sustainability and the SDGs to propose improvements

Here are some statements that demonstrate how a learner in HND Software Development might encounter concepts relating to sustainability, both positively and negatively.

Positive statements

As a software developer, you:

- explore sustainable coding practices that prioritise energy efficiency and minimise resource consumption
- develop software solutions that contribute to environmental sustainability, such as creating applications that promote recycling or monitor energy usage

- understand how software development can support sustainable business practices, like developing systems for optimising supply chains and reducing waste
- collaborate with experts in sustainable development to create software applications that address environmental challenges, such as climate change or conservation efforts
- analyse the positive impact of software solutions in promoting sustainable agriculture, renewable energy, or smart transportation systems

Negative statements

There are challenges that software developers may encounter in developing software solutions, including:

- the potential for energy-intensive computing, which can contribute to increased carbon emissions and environmental harm
- pressure to prioritise functionality and speed over sustainability considerations, resulting in resource-intensive applications with larger environmental footprints
- working with outdated legacy systems that lack sustainability considerations, posing obstacles to implementing eco-friendly software solutions
- resource-intensive gaming or cryptocurrency mining software creating a negative environmental impact due to high energy consumption
- potential for software solutions to overlook social justice aspects, leading to biases or discriminatory outcomes in algorithms, perpetuating inequality rather than promoting sustainable development goals

You should consider how any case study or real-world problem you give to learners has the potential to impact sustainability either negatively or positively. You should encourage learners to consider how they might mitigate any adverse impacts.

You can cover any of the SDGs that are relevant to the subject area.

Find out more about SQA's approach on the NextGen: HN Learning for Sustainability web page. There is an LfS reflective template available in the resources section. You may find it helpful as a starting point for considering how the SDGs are, or could be, embedded in a qualification, unit or assessment.

Grading

Please see the Grading Pack for this qualification for more information on making grade judgements.

Grading in NextGen: HN Qualifications produces a valid and reliable record of a learner's level of achievement across the breadth of the qualification content.

As well as grading the whole qualification, you assess individual units on a pass or fail basis. Each unit has evidence requirements that learners must achieve before you can consider them for whole-qualification grading.

Whole-qualification grade outcomes

Learners who pass NextGen: HN Qualifications receive one of the following grade outcomes for the qualification as a whole:

- Achieved with Distinction
- Achieved with Merit
- Achieved

To determine a learner's whole-qualification grade, you use the grading matrix provided in the Grading Pack to assess and judge their performance across the key aspects of the HND. You must align your judgements with the following whole-qualification grade descriptors.

Whole-qualification grade descriptors

Achieved with Distinction

The learner has achieved an excellent standard across the course content, going significantly beyond meeting the qualification requirements. They showed a comprehensive knowledge and understanding of course concepts and principles, and consistently used them to apply skills to complete high-quality work. They engaged significantly with the process of developing their meta-skills in the context of their HN Qualification.

Achieved with Merit

The learner has achieved a very good standard across the course content, going beyond meeting the qualification requirements. They showed a very good knowledge and understanding of course concepts and principles, and consistently used them to apply skills to complete work of a standard above that expected for an Achieved grade. They actively engaged with the process of developing their meta-skills in the context of their HN Qualification.

Achieved

The learner has achieved a good standard across the course content, credibly meeting the qualification requirements. They showed a good knowledge and understanding of course concepts and principles, and used them to apply skills to complete work of the required standard. They engaged with the process of developing their meta-skills in the context of their HN Qualification.

Approaches to delivery and assessment

You should prioritise delivering the Software Development mandatory unit as early as possible, as it provides the underpinning for many optional units, such as Software Engineering Methods, Event Driven Programming and Application Development for Web. Learners should commence the mandatory project unit (Professional Practice in Software Development) when they have sufficient breadth of knowledge in their optional topics to apply the concepts, principles and skills to the solution of a significantly large and complex software project.

You should consider when you might deliver units in tandem to enhance learning, or where sequential delivery might provide the best learning opportunity. In all the units, learners should realise the software development concepts through practical experiences with real-world problems demanding a software solution. The more realistic these are, the more powerful the learning experience. The increasing availability of virtual cloud infrastructures from technology vendors offers scope to challenge learners to apply their knowledge and skills to unfamiliar contexts.

Many of the units are suited to assessment in the form of assignments and/or projects. In such cases the product evidence is specified clearly, and you should use this information to plan work that enables learners to meet the evidence requirements. The case studies and assignments used for the purpose of generating assessment evidence should be as real-world as possible, while considering the SCQF level of the unit. Where program code is required, centres should ensure that learners have sufficient access to programming resources such as an integrated development environment (IDE) or notebook-style environments.

Sequencing or integrating units

As indicated above, some units would benefit from co-delivery to enhance the learning experience. Typical examples of these could be:

- Application Development for Web and Database Design and Development
- Code Security and Event Driven Programming
- Software Development and Software Engineering Methods

Alternatively, the units could follow the process of software development from establishing user requirements through software design and development to testing and implementation.

Additional guidance on integrated or holistic assessment

Holistic or integrated assessment focuses on assessing a number of outcomes in a unit together, or in some cases, assessing the unit as a whole, rather than by outcome. When assessing a unit of competence holistically, the assessment activities integrate a number of aspects of the competence. Holistic or integrated assessment can reduce the time spent on assessment and can promote greater equity in the assessment process.

When developing or revising a NextGen: HN Qualification, SQA works with a development team to devise an appropriate assessment strategy that accommodates holistic or integrated assessment. However, the practice of integrating units for the purposes of learning and teaching is a centre-led activity.

Units are designed to facilitate holistic or integrated assessment approaches that prevent large, unwieldy assessments.

Sometimes more than one piece of evidence is needed for a unit. For example, if a unit is about creating code, a learner would need to produce evidence of knowledge (following the software development processes) and product (a working piece of code).

Evidence requirements must do what they say: specify requirements for evidence of learner competence in the unit. The evidence must be of sufficient quality for an assessor or verifier to judge that the learner has achieved the unit.

Remediation and re-assessment in Next Generation Higher National Qualifications

Remediation

Remediation allows an assessor to clarify learners’ responses, either by requiring a written amendment or by oral questioning, where there is a minor shortfall or omission in evidence requirements. In either case, the assessor must formally note such instances, in writing or as a recording, and make them available to the internal and external verifier.

Remediation is not permitted for closed-book assessments.

The size and structure of the larger NextGen: HN units should mean that the assessor or lecturer is close enough to ongoing assessment activity in project-based units to identify the requirement for remediation as it occurs.

Re-assessment

We must give learners who fail the unit a re-assessment opportunity or, in exceptional circumstances, two re-assessment opportunities. Where we have introduced larger units to the framework, we expect instances of re-assessment to be minimal, due to the approach to assessment and remediation. Where re-assessment is required in a project-based unit, a substantially different project must be used.

Resource requirements

To ensure that learners can develop their knowledge and skills across the mandatory and optional units in the framework, you should consider the following resources:

- Computers: provide each learner with a dedicated computer workstation for programming tasks. Ensure they meet the minimum system requirements for software development tools and IDEs that you intend to use in the course.

- Programming tools and IDEs: install one or more of the IDEs in common use.
- Operating systems: install operating systems that expose learners to different environments, including Windows and Linux.
- Version control systems: introduce learners to version control and collaboration tools.
- Agile project management tools: familiarise learners with project management methodologies and tools.
- Software engineering methods: provide learners with resources and materials covering software engineering principles, including:
    - online tutorials and courses on software engineering best practices
    - case studies and examples of software engineering projects
- Database management systems: install and configure one or more popular database systems.
- Human-Computer Interaction (HCI) tools: demonstrate user interface and user experience design tools.
- Event driven programming: provide learners with resources and materials on event driven programming, including:
    - examples and tutorials on event driven programming in different languages
- Data science libraries and tools: introduce learners to data science programming with libraries and tools including access to notebook style systems.
- Web application development: teach learners web application development using:
    - HTML and CSS
    - JavaScript frameworks
    - backend frameworks
- Code security tools and practices: educate learners about code security and provide resources for static code analysis, and security testing.
- Collaboration and communication tools: set up tools to facilitate collaboration and communication among learners.
- Practice projects: prepare a set of practice projects and assignments for learners to apply their knowledge and skills.

- Guest speakers and mentors: invite industry professionals to give talks or act as mentors to provide real-world insights and guidance.

Information for centres

Equality and inclusion

The units in this HND are designed to be as fair and as accessible as possible with no unnecessary barriers to learning or assessment.

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

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

Internal and external verification

You must make sure all instruments of assessment you use in this qualification are internally verified according to your centre's policies and SQA's guidelines.

SQA carries out external verification to ensure that internal assessment meets the national guidelines for this qualification.

More information on internal and external verification is available in SQA's Guide to Assessment and in Next Generation: Higher National Quality Assurance — Guidance for Centres.

Glossary

SQA credits: 1 SQA credit equals 8 SCQF credit points.

SQA credit value indicates the contribution the unit makes to an SQA qualification. An SQA credit value of 1 represents approximately 40 hours of learning, teaching and assessment.

SCQF: the Scottish Credit and Qualifications Framework (SCQF) is Scotland's national framework for describing qualifications. We use SCQF terminology in this guide to refer to credits and levels. For more information on the SCQF, visit the SCQF website.

SCQF credit points indicate the amount of learning required to complete a qualification. NextGen HNCs and HNDs are worth 120 SCQF credit points.

SCQF levels indicate how hard the qualification is to achieve. The SCQF covers 12 levels of learning. NextGen HNCs are at SCQF level 7 and NextGen HNDs are at SCQF level 8.

Information for learners

HND Software Development

This information explains:

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

Qualification information

HND Software Development provides you with the opportunity to gain the knowledge, understanding and skills you need to pursue a career in software development and/or to progress to higher levels of qualification. The demand for software developers remains high and is expected to grow in the coming years as businesses and organisations increasingly adopt digital solutions. Software development is constantly evolving, with new technologies and techniques that extend the range of digital solutions to improve business efficiency.

This qualification prepares you for a career in software development, with the skills and knowledge you need to succeed in the workplace. You can develop your professional skills and apply them to diverse challenges. This is ideal preparation for advancement to further study, including qualifications at degree level.

Before you begin, you should have a basic understanding of computer programming and be able to design, code and test a program to a given brief. You should also understand the fundamental concepts in computer science, including the application of computing to solve a range of problems. You should be a confident user of software tools to perform tasks related to managing your studies and reporting evidence. You can evidence this by having an HNC qualification in computing, or equivalent at SCQF level 7.

During the course, you experience modern approaches to the design, development and testing of software applications using a range of programming languages and tools. You use structured methodologies to discover and analyse user requirements and design software solutions that meet those requirements. As team working is so important in software development, you work collaboratively with others to develop software applications, using version control and other tools to manage the development process. You evaluate software development methodologies and choose appropriate approaches for project work on the course, including code quality, testing, and documentation.

To achieve the qualification, you must pass two mandatory units (achieving 7 SQA credits) and pass sufficient optional units to achieve another 8 SQA credits. One of the mandatory units is an extensive collaborative project to design and develop a software application, using project management and other software development processes, including version control, testing, and technical documentation. This project also contributes to the grading of your HND. The optional units give you the chance to develop an area of specialism, such as software engineering or web applications.

HND Software Development helps you achieve the following specific aims:

1. Develop understanding of the processes, paradigms and concepts that underpin the development of software solutions, including object orientation.
2. Develop a range of specialist skills in software development, including programming and databases.
3. Develop competence in applying professional skills, techniques and practices associated with contemporary software development.
4. Develop understanding of current trends in software development, such as secure development and operations, and how they are applied for the benefit of society and the economy.
5. Provide practical experience and professional behaviours in applying theory to a range of real-world contexts.
6. Prepare learners for employment in the design, development, testing and implementation of software solutions to problems.

7. Develop computational thinking, maker mentality, pattern recognition, deconstruction, logical thinking, teamworking and collaboration, and synthesis and analysis skills.
8. Gain understanding of the ethical, social and legal issues associated with software development, including those relating to diversity, inclusion and sustainability.
9. Prepare for progression to further studies in software development, or related disciplines, at SCQF level 9.

You are assessed through a variety of approaches that reflect modern practice in assessment, including the use of projects, assignments and investigations, as well as question papers. You should use a range of media to present evidence, such as video, audio, web pages and social media platforms.

Your final qualification (provided all units are passed) is graded in a manner that reflects the quality of your work over the course. Project work that you have completed in the units of the course is graded according to a set of criteria that relate to the competences expected of a software development practitioner.

Throughout HND Software Development you also develop your personal meta-skills while studying industry and sector-specific content. Meta-skills are higher-order skills that support the development of other skills and promote success in any context. They enable you to respond to professional challenges and opportunities by reflecting on, developing, applying and adapting industry skills and sector knowledge. These
meta-skills are grouped into three categories: self-management, social intelligence, and innovation.

You also develop your knowledge, skills and values related to the challenges of sustainability. You are encouraged to ask questions, analyse, think critically and work with others to make positive contributions to a sustainable future.

Successful completion of HND Software Development opens up career opportunities as a junior software developer. It also prepares you for progression to a degree-level qualification in software development or further study in related computing topics.

Administrative information

Published: October 2025 (version 1.0)

History of changes

Version	Description of change	Date

Please check SQA's website to ensure you are using the most up-to-date version of this unit.

If a unit is revised:

- no new centres can be approved to offer the previous version of the unit
- centres should only enter learners for the previous version of the unit if they can complete it before its finish date

For more information on NextGen: HN Qualifications please visit the NextGen: HN web page.