Next Generation Higher National Educator Guide

Higher National Diploma in Software Development

Qualification code GV22 48

Valid from session 2024 to 2025

Prototype educator guide for use in pilot delivery only (version 0.1) August 2024

This guide provides detailed information about the qualification to ensure consistent and transparent assessment year on year.

This guide is for assessors and lecturers and contains all the mandatory information you need to deliver and assess the qualification.

This edition: August 2024 (version 0.1)

Contents

Qualification overview

Introduction

This guide:

- assists centres to implement, deliver, and manage the qualification
- provides a guide for new staff involved in offering the qualification
- informs course managers, teaching staff, assessors, learners, employers, and higher education institutions of the aims and purpose of the qualification
- provides details about the range of learners that the qualification is suitable for and the progression opportunities

Purpose of the qualification

HND Software Development addresses a specialist area within 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.

The structure and aims of the HND are described in the following sections.

Structure

Higher National Diplomas (HNDs) are designed at SCQF level 8 and consist of 120 SCQF credit points. HNDs must incorporate at least 80 credit points (10 credits) at SCQF level 8.

HNDs contain 15 credits that can be used flexibly to increase opportunities for learners returning to education. Refer to the 'Meta-skills' section of this guide for more information.

HND Software Development meets the Next Generation Higher National (NextGen: HN) Design principles by incorporating a mandatory project unit (Professional Practice in Software Development at SCQF level 8) with an SQA credit value of 4.

Framework

The HND Software Development is made up of the following two mandatory units (7 SQA credits) and 8 SQA credits from the table of optional units (15 SQA credits in total).

Mandatory units

Unit code	Unit title	SQA credit	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 credit	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

The HND Software Development provides learners with knowledge and skills in the development, testing and deployment of software solutions that satisfy requirements.

This includes the use of structured methodologies to discover and analyse client requirements and design appropriate software solutions. The HND covers modern approaches to the design, development and testing of software applications using a range of programming languages and tools.

The HND includes an extensive collaborative project for learners to design and develop a software application using project management and other software development processes, including version control, testing, and technical documentation.

General aims

The general aims of the qualification are to:

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 develop professionalism and meta-skills to support lifelong professional learning and development
5 enable progression through the Scottish Credit and Qualification Framework (SCQF), including progression to degree-level study

Specific aims

The specific aims of the qualification are to:

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, synthesis and analysis skills
8 gain understanding of the ethical, social and legal issues associated with the field of 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

Qualification structure

The tables that follow show how the mandatory and optional units contribute to the general and specific aims of the HND.

The HND structure mirrors that set out in the NextGen: HN Design Rules. There are two mandatory units, of which one is the project unit (Professional Practice in Software Development, 4 SQA credits) and the other a unit that covers the concepts that are essential to the practice of software development (Software Development, 3 SQA credits). The project unit provides opportunity for learners to integrate the knowledge and skills covered in the Software Development unit and/or explore in depth a software development topic from the set of optional units.

The options group enables learners to develop an area of specialism such as software engineering or web applications. This HND is not a minimum viable product, but a fully fleshed out qualification in software development and its associated areas of specialism.

Who is this qualification for?

This HND is suitable for learners who wish 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.

Recommended entry	Progression
Entry to this qualification is at the discretion of your centre. Learners would benefit from having attained the skills, knowledge, and understanding required by one or more of the following or equivalent qualifications and/or experience: ♦ HNC Computing (NextGen: HN) (GT6G 47) ♦ HNC Computing (GF3E 15)	♦ other qualifications in computing or related areas ♦ degree-level study in software development ♦ further study, employment and/or training in software development

Recognising prior learning

SQA recognises that learners gain knowledge and skills through formal, non-formal, and informal learning contexts.

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

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

Articulation and/or progression

The 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.

Professional recognition

There is no professional recognition associated with this HND.

Credit transfer arrangements

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

Core Skills entry profile

The Core Skills entry profile provides a summary of the assessment activities that demonstrate the SCQF level of this qualification. This information can help identify learners that need additional support or those who should take an alternative level or learning programme.

Core Skill	Recommended SCQF entry profile	Associated assessment activities
Communication	Level 6	♦ performing a complex task ♦ making a sustained and coherent argument on a complex topic ♦ making a formal presentation that collates, analyses, and presents conclusions on findings ♦ summarising and evaluating research reports ♦ producing a well-structured report of a complex, practical investigation
Numeracy	Level 6	♦ calculations involving complex data (for example, financial) ♦ using formulae to calculate values ♦ calculating data volumes or processing time for software applications
Information and communications technology (ICT)	Level 6	♦ searching the internet for examples and solutions to coding challenges ♦ using a software package to create documentation or develop code ♦ recording a video on a mobile device and uploading to video streaming services

Core Skill	Recommended SCQF entry profile	Associated assessment activities
Problem solving	Level 6	♦ analysing the significance and relevance of variables in a situation ♦ summarising, explaining or drawing conclusions ♦ identifying and ensuring sufficient resources are available ♦ deciding how a task will be managed and carrying it out ♦ analysing the effectiveness of all aspects of a problem-solving strategy ♦ identifying and gathering appropriate evidence to support an evaluation
Working with others	Level 6	♦ analysing requirements of roles and relationships between them ♦ negotiating working methods with team members ♦ using reflection and feedback received to identify learning objectives

Meta-skills

Meta-skills are higher-order skills that support the development of other skills and promote success in any context. They enable learners to respond to professional challenges and opportunities by reflecting on, developing, applying and adapting industry skills and sector knowledge.

Our new Higher National Qualifications are developed with meta-skills at their core. Meta-skills complement the industry and sector-specific content of the qualifications. They provide a framework for learners to complete personal development aligned to professional practices.

Throughout the qualifications, learners develop meta-skills while studying industry and sector-specific content. You can integrate meta-skills into contextualised teaching activities and include them in integrated and holistic assessment approaches.

The 21st century skills and meta-skills learning, teaching and assessment model focuses on how we can use skills to respond to societal, economic and industry drivers and change. Meta-skills frameworks vary, but they share an approach that emphasises individualistic, context-based skills development with reflective practice and localised definitions.

Skills Development Scotland developed a model of meta-skills in response to the concept of Industry 4.0 (or the 'fourth industrial revolution'). In this model, they identify 12 meta-skills that help learners adapt to changes to industry, job roles and society expected as a result of technological advances and global trends. Developing these meta-skills supports learners as they prepare for a constantly evolving future.

The 12 meta-skills are grouped into three categories: self-management, social intelligence, and innovation.

Self-management	Social intelligence	Innovation
Focusing	Communicating	Curiosity
Integrity	Feeling	Creativity
Adapting	Collaborating	Sense-making
Initiative	Leading	Critical thinking

Adapted from: Skills 4.0: a skills model to drive Scotland's future, Centre for Work-based Learning in Scotland, (2018).

You should:

- make learners aware that meta-skills are generic and transferable across many different contexts
- support learners to focus on the meta-skills that they find most relevant by encouraging an individualised, active learning approach that relates to the industry and sector contexts of the qualification

- help learners to understand key meta-skills for their industry or sector and any other personally important meta-skills, and set development goals for these
- encourage learners to focus on reflective practice

None of the meta-skills are mandatory.

Learning and teaching

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. Within 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 they can be applied. 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 UN 2030 Agenda for Sustainable Development, adopted by the UK in 2015, has shaped the development of internal and national sustainability policy. It sets out the United Nations 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.

In line with this, SQA is committed to incorporating the skills, knowledge, understanding and values of LfS within all new and revised qualifications.

LfS combines:

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

ESD is the internationally used term for sustainability education. LfS has a broader remit; however, the terms are largely interchangeable. ESD tends to be used by colleges and universities, while LfS is usually used in schools. 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.

LfS is designed to nurture a generation of learners who know the value of the natural world and are committed to the principles of social justice, human rights, global citizenship, democratic participation and living within the ecological limits of the planet. It aims to respond to global challenges by developing learners' skills, knowledge, understanding and values relating to sustainability so they can interact with the world in a socially responsible way.

LfS is more than the sum of its parts; it is about building learners' capacity to deal with the unpredictable challenges facing our rapidly changing world. It encourages transformational change through learning, by which learners are able to critically analyse, communicate and collaborate on complex social, environmental and economic challenges. This gives learners increased confidence, opportunities to develop a range of meta-skills, and enhanced motivation and readiness to learn.

Learning for Sustainability in Next Generation Higher National Qualifications

Next Generation Higher National (NextGen: HN) Qualifications have been developed with sustainability as a core component.

All NextGen: HN learners should exit their qualification with:

- a general understanding of sustainability and the SDGs
- an understanding of subject-specific sustainability issues, how these relate to the SDGs, and potential improvements
- the confidence to apply their knowledge and skills in the next stage of their lives

Central to these aims is a need for familiarity with both the SDGs and the concept of sustainability (which is the need to ensure a balance between economic growth, environmental stewardship and social well-being). Knowledge and understanding of current industry practices and behaviours, and consideration of how these could be made more sustainable and contribute towards the SDGs, are integral in developing young people to be responsible and empowered citizens who are able to contribute to building a socially just, sustainable and equitable society.

With this in mind, sustainability is embedded as an outcome in the Professional Practice in Software Development project unit.

By completing this outcome, learners develop skills, including the abilities to:

- 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

Any of the SDGs can be covered; there are none that are mandatory.

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

On the other hand, 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.

Grading

Please see the Grading Pack for this qualification for more information on grading.

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

You assess and judge each learner's performance across the key aspects of the qualification to determine their whole qualification grade. You must align judgements with the whole qualification grade descriptors, which are:

Achieved with Distinction

This candidate consistently demonstrates outstanding knowledge, understanding and application of skills. Thinking and working independently to an exceptional standard, they apply excellent judgement and creative problem-solving skills. They achieve or exceed agreed aims by confidently applying an extensive range of meta-skills and working very effectively with colleagues and peers.

Achieved with Merit

This candidate demonstrates an excellent level of knowledge, understanding and application of skills. Thinking and working independently to a high standard, they demonstrate good judgement and effective problem-solving skills. They achieve agreed aims by applying a broad range of meta-skills and working effectively with colleagues and peers.

Achieved

This candidate demonstrates thorough knowledge, understanding and application of skills. They think and work independently and use their judgement to find solutions to problems. They achieve agreed aims by applying a range of meta-skills and working well with colleagues and peers.

Successful learners receive their grade, along with the grade descriptor text, on their commemorative certificate.

In addition, 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.

You make judgements about learners' quality of assessment evidence using a grading matrix based on important criteria in the qualification.

Grading and meta-skills

Meta-skills are a key part of the NextGen: HN qualifications and learners develop them throughout the qualification. Competence in individual meta-skills is not assessed or graded. For example, the qualification does not judge the quality of learners' feeling or creativity, or their specific progress in any given meta-skill. Rather, it is the process of development the learner goes through that contributes to the whole qualification judgement. This means learners should provide evidence of planning, developing and reflecting on their meta-skills. The grading matrix includes criteria on meta-skills, which you should use to support this judgement.

How the qualification meets employer needs

This HND is designed in collaboration with employers to meet the sector need. The following tables show how the HND can benefit employers by producing learners with the necessary skill set.

They show how units map to the aims of the HND; how the units map to National Occupational Standards for IT Professionals; the significant opportunities that the HND provides for learners to develop more generic skills and meta-skills; and the assessment strategy for the HND.

Mapping qualification aims to units

General aims

Key: Aim is relevant to unit (X) Aim is not relevant to unit (—)

Unit code	Unit title	General aims 1	General aims 2	General aims 3	General aims 4	General aims 5
J7D9 48	Software Development	X	X	X	X	X
J7EE 48	Professional Practice in Software Development	X	X	X	X	X
J7E1 48	Application Development for Web	—	X	X	X	X
J7EB 48	Code Security	—	X	X	—	X
J7DV 48	Database Design and Development	—	X	X	—	X
J897 48	DevOps Principles and Practice	X	X	X	X	X
J691 47	Emerging Technologies and Experiences	X	X	—	—	X
J7EC 48	Event Driven Programming	—	X	X	—	X
J7ED 48	Human Computer Interface	—	X	X	X	X
J7EG 48	Programming for Data	—	X	X	—	X
J7EF 48	Software Engineering Methods	X	X	X	X	X
J7D9 48	Software Development	X	X	X	X	X

Specific aims

Key: Aim is relevant to unit (X) Aim is not relevant to unit (—)

Unit code	Unit title	Specific aims 1	Specific aims 2	Specific aims 3	Specific aims 4	Specific aims 5	Specific aims 6	Specific aims 7	Specific aims 8	Specific aims 9
J7D9 48	Software Development	X	X	X	X	X	X	—	X	X
J7EE 48	Professional Practice in Software Development	X	X	X	X	X	X	X	X	X
J7E1 48	Application Development for Web	—	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	—	X
J691 47	Emerging Technologies and Experiences	—	—	—	X	—	—	—	X	X
J897 48	DevOps Principles and Practice	X	—	X	X	X	X	X	X	X
J7EC 48	Event Driven Programming	X	X	X	—	X	X	X	—	X
J7ED 48	Human Computer Interface	X	X	X	X	X	X	—	X	X
J7EG 48	Programming for Data	X	X	X	—	X	X	X	X	X
J7EF 48	Software Engineering Methods	X	X	X	X	X	X	X	X	X

Mapping National Occupational Standards (NOS) and/or trade or professional body requirements to units

Key: Aim is relevant to unit (X) Aim is not relevant to unit (—)

Unit code	Unit title	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	6.3	6.4	8.1	10.2	10.3	10.6
J7D9 48	Software Development	X	X	X	—	X	X	X	—	X	X	X	—	X	X	—	—	—	—	—	—
J7EE 48	Professional Practice in Software Development	X	X	X	X	X	X	X	X	X	X	X	X	X	X	—	—	—	X	X	X
J7E1 48	Application Development for Web	X	—	—	—	—	—	—	—	—	X	X	—	—	X	—	X	X	X	X	X
J7EB 48	Code Security	—	—	—	—	—	—	—	—	X	X	X	X	—	—	X	X	X	—	—	—
J7DV 48	Database Design and Development	X	X	X	X	—	—	X	—	—	X	X	X	—	—	—	—	X	—	—	X
J897 48	DevOps Principles and Practice	X	X	X	X	X	X	X	X	—	—	—	X	—	X	—	—	—	—	—	X
J691 47	Emerging Technologies and Experiences	X	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	X
J7EC 48	Event Driven Programming	—	—	—	—	—	X	X	X	—	X	X	X	—	—	—	X	—	—	—	X
J7ED 48	Human Computer Interface	X	X	—	—	X	—	—	X	X	X	X	—	X	X	—	—	—	—	—	—

Unit code	Unit title	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	6.3	6.4	8.1	10.2	10.3	10.6
J7EG 48	Programming for Data	—	—	—	X	—	—	X	—	—	X	X	—	X	X	—	X	—	—	—	X
J7EF 48	Software Engineering Methods	X	X	X	X	X	X	X	X	—	—	—	X	—	X	—	—	—	—	—	X

Mapping opportunities to develop meta-skills across the qualification

Self-management

Unit code	Unit title	Meta-skills
J7EE 48	Professional Practice in Software Development	♦ Focusing ♦ Integrity ♦ Adapting ♦ Initiative
J7EG 48	Programming for Data	♦ Focusing ♦ Adapting
J7EC 48	Event-Driven Programming	♦ Focusing ♦ Adapting
J7EB 48	Code Security	♦ Focusing ♦ Integrity
J7EF 48	Software Engineering Methods	♦ Focusing ♦ Initiative
J7D9 48	Software Development	♦ Focusing ♦ Integrity ♦ Initiative
J7ED 48	Human Computer Interface	♦ Focusing ♦ Integrity ♦ Adapting ♦ Initiative
J7DV 48	Database Design & Development	♦ Focusing ♦ Initiative
J7E1 48	Application Development for Web	♦ Adapting ♦ Initiative
J897 48	DevOps Principles and Practice	♦ Focusing ♦ Adapting ♦ Initiative

Social intelligence

Unit code	Unit title	Meta-skills
J7EE 48	Professional Practice in Software Development	♦ Communication ♦ Feeling ♦ Collaborating ♦ Leading
J7EC 48	Event-Driven Programming	♦ Communication
J7ED 48	Human Computer Interface	♦ Communication ♦ Feeling ♦ Collaborating
J7E1 48	Application Development for Web	♦ Communication ♦ Collaborating ♦ Leading
J7D9 48	Software Development	♦ Communication ♦ Feeling ♦ Collaborating ♦ Leading
J7EG 48	Programming for Data	♦ Communication
J7EF 48	Software Engineering Methods	♦ Collaborating ♦ Leading
J7DV 48	Database Design and Development	♦ Collaborating ♦ Leading
J897 48	DevOps Principles and Practice	♦ Communication ♦ Feeling ♦ Collaborating ♦ Leading

Innovation

Unit code	Unit title	Meta-skills
J7EE 48	Professional Practice in Software Development	♦ Curiosity ♦ Creativity ♦ Sense-making ♦ Critical thinking
J7EC 48	Event-Driven Programming	♦ Creativity ♦ Sense-making ♦ Critical thinking
J7ED 48	Human Computer Interface	♦ Curiosity ♦ Creativity ♦ Sense-making ♦ Critical thinking
J7E1 48	Application Development for Web	♦ Curiosity ♦ Sense-making ♦ Critical thinking
J7D9 48	Software Development	♦ Curiosity ♦ Creativity ♦ Sense-making ♦ Critical thinking
J7EG 48	Programming for Data	♦ Creativity ♦ Sense-making ♦ Critical thinking
J691 47	Emerging Technologies and Experiences	♦ Sense-making ♦ Critical thinking
J7EF 48	Software Engineering Methods	♦ Sense-making ♦ Critical thinking
J7DV 48	Database Design and Development	♦ Creativity ♦ Sense-making ♦ Critical thinking
J897 48	DevOps Principles and Practice	♦ Sense-making ♦ Critical thinking

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.
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.

Unit code	Unit title	Assessment method
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 the following evidence: — 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
J7DV 48	Database Design and Development	♦ Product evidence: The product evidence consists of: — a normalised relational database, appropriately populated, from a given specification — a set of queries that learners use to access, analyse and manage the data — a NoSQL database, appropriately populated — a 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
J69147	Emerging Technologies and Experiences	♦ Knowledge evidence: In any form of media, learners demonstrate knowledge relating to each outcome.

Unit code	Unit title	Assessment method
J7EC48	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
J7ED48	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.
J7EG48	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
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, covering the gamut of methods and processes from establishing software requirements to the design and validation of a software solution.

Approaches to delivery and assessment

You should prioritise the Software Development mandatory unit for delivery 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 in this HND, 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 is. 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 clearly specified, 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 taking into account 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 the whole unit, rather than specific outcomes. 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 Higher National 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 instruments of assessment.

Sometimes more than one piece of evidence is needed for a unit. For example, if a unit is about building a wall, a learner would need to produce evidence of performance (following the correct procedures and processes when building the wall) and product (a completed wall).

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

Assessing project units

The Professional Practice in Software Development unit comprises a software project that is sufficiently large or complex to require analysis and solution by a project team. You should organise your group into teams of three to five learners, taking into account the preferences expressed by learners in relation to the available projects and your judgement of an appropriate composition for each team, which should vary in terms of age, gender, ability and other relevant characteristics.

You should have a range of software development projects that are suitable to enable learners to demonstrate the outcomes of the project unit. You should select these to be as authentic (real-world) as possible and aligned to the level of study required. The project should allow learners to draw on the range of outcomes in their HN qualification and make it possible for them to develop their meta-skills while engaging in the project. To standardise the learner experience and level of demand, you should use a template to describe the objectives and required outcomes for each project.

You should:

- approve projects
- assign learners to teams
- ensure each team allocates roles at the beginning of the project
- receive briefings from each team on progress against milestones
- observe and record the contributions of individual learners to teamwork and solution
- arrange for the presentation of the project outcomes and solution
- intervene to guide and support learners when circumstances require it

You should approach your role with a light touch, leaving each team to make their own decisions about a project methodology, as well as roles, timelines, resource allocations and meetings. We recommend that each learner leads a specific part of the project (such as lead analyst or lead software designer).

The product evidence is a software solution to a problem whose requirements must be sufficiently complex to require formal project management and development approaches. Each project team must collectively produce the following items 1 to 5. Each learner must individually produce item 6. Learners must provide the following product evidence:

1 the project plan based on requirements gathering and analysis
2 the software solution to the problem
3 a project report (including project evaluation)
4 documentation of the product (technical and user)
5 presentation of the project outcomes and demonstration of solution
6 personal statement

The software product must provide a working solution that meets the requirements set out in the project plan. The learner's personal statements must:

- describe their specific role or roles in the project team
- include a self-evaluation of their contribution to the development of the digital product

The performance evidence is in two parts:

- a record of the learner's professional behaviours and contributions to the project; this may be in the form of a checklist
- a recording of performance in delivering the presentation

Each learner must make an identifiable contribution to the final presentation of the project outcomes and the demonstration of the solution; the contribution can be in any form that is appropriate and agreed with the assessor.

Opportunities for e-assessment

If you want to use e-assessment, you must ensure that you apply the national standard to all evidence and that conditions of assessment (as specified in the evidence requirements) are met, regardless of the mode of gathering evidence.

Learners need to use software products for most of the assessments that require product evidence. This results in a high proportion of digital evidence, which can be gathered in an e-portfolio.

Where knowledge evidence is required, this can often be generated in digital form such as text documents, slide decks, audio files and video recordings. Where testing is used, then online approaches to tests should be adopted where practicable.

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 qualification are 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.

Internal and external verification

All instruments of assessment used in this qualification should be 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.

Further information on internal and external verification is available in SQA's Guide to Assessment.

Glossary

SQA credit value: the credit value allocated to a unit gives an indication of the contribution the unit makes to an SQA qualification. An SQA credit value of 1 represents approximately 40 hours of programmed learning, teaching, and assessment.

SCQF: the Scottish Credit and Qualifications Framework (SCQF) provides the national common framework for describing programmes of learning and qualifications in Scotland. SCQF terminology is used throughout this guide to refer to credits and levels. For further information on the SCQF, visit the SCQF website.

SCQF credit points: SCQF credit points provide a way of describing and comparing the amount of learning required to complete a qualification at a given level of the framework. 1 National Unit credit is equivalent to 6 SCQF credit points. 1 National Unit credit at Advanced Higher and 1 SQA Advanced unit credit (irrespective of level) is equivalent to 8 SCQF credit points.

SCQF levels: the level a qualification is assigned in the framework is an indication of how hard it is to achieve. The SCQF covers 12 levels of learning. SQA Advanced Certificates and SQA Advanced Diplomas are available at SCQF levels 7 and 8, respectively. SQA Advanced units are usually at levels 6 to 9 and graded units at level 7 and 8. National Qualification group awards are available at SCQF levels 2 to 6 and are usually made up of National Units, which are available from SCQF levels 2 to 7.

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 will need to do during the qualification
- opportunities for further learning and employment

Qualification information

The HND Software Development provides you with the opportunity to gain the knowledge, understanding and skills that enable you 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 HND prepares you for a career in software development, with the skills and knowledge you need to succeed in the workplace. You have opportunities to 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 this 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.

The specific aims of the HND Software Development are to enable you to:

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 the application of 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 their application for the benefit of society and the economy
5 provide practical experience and professional behaviours in the application of theory to a range of real-world contexts
6 prepare 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, synthesis and analysis skills
8 gain understanding of the ethical, social and legal issues associated with the field of 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 are encouraged to use a range of media to present evidence, such as video, audio, web pages and social media platforms.

Your final award (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 the 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.

In line with government policy, 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 the 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: August 2024 (version 0.1)

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 guide, and check SQA's APS Navigator to ensure you are using the most up-to-date qualification structure.

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 further information on SQA's Next Generation Higher National Qualifications please contact nextgen@sqa.org.uk.