Next Generation Higher National Unit Grading Pack

Higher National Diploma Computer Science

Qualification code: GV20 48

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

This grading pack provides information about the process of grading the Higher National Diploma (HND) Computer Science. It is for lecturers and assessors, and contains all the mandatory information you need to grade the HND.

You must read it alongside the Educator Guide.

Published: October 2025 (version 1.0)

Contents

Approach to grading

Grading in Next Generation: Higher National (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 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.

What the whole-qualification grade descriptors do and how they are used

The whole-qualification grade descriptors outline the skills, knowledge and understanding a learner needs to show across the whole qualification to achieve that specific grade. They align with the Scottish Credit and Qualifications Framework (SCQF) level descriptors.

NextGen: HND qualifications are at SCQF level 8. Learners who complete a NextGen: HND can:

- convey an insightful understanding of the subject’s core theories, concepts and principles, along with its scope and defining features
- apply skills, knowledge and understanding of the subject in relevant practical and professional contexts, showing some specialist knowledge and using a range of relevant techniques and materials
- describe and explain significant topical issues and specific areas of interest

- exercise autonomy and initiative in carrying out activities, and have developed their professional practice and behaviours relevant to the context of the qualification
- formulate and critically evaluate evidence-based responses to issues in the context of the subject area, appropriately applying research and academic processes

Please use this information, as well as the whole-qualification grade descriptors, to help you understand the standard at which learners should be assessed and graded.

Higher education institutes (HEIs) can use the grade descriptors to set admissions requirements, and employers can use them to help make decisions during a recruitment process.

SQA’s quality assurance teams use the grade descriptors and the grading matrix to ensure that grades awarded in a particular NextGen: HN Qualification are at a consistent national standard, regardless of the setting in which they are achieved.

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

Using the grading matrix

You must use the grading matrix to judge the learner’s whole-qualification grade. You can use the grading matrix at any time, but you only make a whole-qualification grading judgement when you are confident the learner has met all the evidence requirements of all the required units.

The criteria in the grading matrix reflect the knowledge, skills and qualities HEIs and employers can expect of a learner who has completed the qualification. These criteria align with the overall purpose of the qualification and remain the same for its duration.

Each criterion has sector-specific descriptors of a typical learner’s performance standard, aligned to the whole-qualification grade outcomes of Achieved, Achieved with Merit and Achieved with Distinction. These descriptors describe the standard a learner of that whole-qualification grade is expected to show.

The guidance accompanying each criterion can include, but is not limited to, information on:

- relevant types of assessment that may produce useful or meaningful evidence for judging that criterion
- mapping to content that is particularly relevant to that criterion
- mapping to meta-skills

This guidance may be updated over time.

When you make your final grading judgement, you must use a ‘best fit’ approach based on the learner’s achievement across the grading matrix. This may be straightforward — for example, if the learner’s evidence shows a consistent standard across the grading matrix criteria. If it is not straightforward, you must make a ‘best fit’ judgement — for example, if a learner shows a mix of standards across the grading matrix criteria, with no clear pattern. The criteria may not always have equal value. You can decide some are more important to the final grade than others.

Meta-skills

Meta-skills are a key part of NextGen: HN Qualifications and learners can develop them throughout the qualification. A learner’s engagement with developing their own meta-skills contributes to their qualification grade. You do not assess or grade competence or progress in individual meta-skills — for example, by judging the quality of a learner’s feeling or creativity. Instead, you look at the process of development learners go through. This means learners need to provide evidence of planning, developing and reflecting on their meta-skills.

If qualification content also contributes to meta-skills development, it contributes to a learner’s whole-qualification grading through the grading matrix approach.

Learning for Sustainability

Learning for Sustainability does not contribute to a learner’s whole qualification grade.

The exception is where Learning for Sustainability content is part of the qualification content. In which case the Learning for Sustainability content will contribute to a learner's whole-qualification grade through the grading matrix.

Grading matrix

Criterion 1 descriptors

Criterion 1	Achieved	Merit	Distinction
Demonstrates knowledge and understanding of computer systems, how they are organised and how they function	The learner: • provides a clear explanation of computer systems architectures • describes in sufficient detail the hardware and software components of a computer system • writes basic assembly code that demonstrates understanding of machine instructions, memory addressing and control flow • correctly installs and configures an operating system with basic settings	The learner: • provides a clear and detailed explanation of computer systems architectures • provides a clear and detailed description of the hardware and software components of a computer system, including their inter-relationships • writes assembly code that solves basic problems and demonstrates efficient use of resources • correctly installs and configures an operating system, including advanced settings and customisation	The learner: • provides a comprehensive explanation of computer systems architectures and how they are optimised • provides comprehensive descriptions of micro-architecture, instruction sets, and other advanced system concepts • writes complex assembly code that demonstrates a sound grasp of the language and uses optimisation techniques • expertly installs, configures and customises an operating system, with detailed documentation

Criterion 1	Achieved	Merit	Distinction
Demonstrates knowledge and understanding of computer systems, how they are organised and how they function (continued)	The learner: • provides clear explanations for a range of data structures and their use cases	The learner: • provides a clear and detailed explanation of a wide range of data structures	The learner: • provides comprehensive and insightful explanations of a wide range of data structures, including their implementation details

Criterion 1 guidance

This competence relates to the professional knowledge and skills that a computer science practitioner requires. This competence can be evidenced in the following units:

Computer Science (J68M 48)

Knowledge evidence relating to outcomes 1,3,4 and 6 that covers:

- computer systems architecture
- discrete maths
- principles of machine learning
- ethical dilemmas in computer science

Product evidence of one or more assembly language programs to demonstrate learner’s practical competency in low-level programming, that includes the design and implementation of a normalised database.

Programming Paradigms (J7DY 48)

Explanations of imperative and declarative paradigms, with several examples of each, including:

- key milestones
- comparison of the following paradigms:
  - low-level
  - procedural
  - object-oriented
  - event-driven
  - functional
  - logic languages

Applied Mathematics for Computing (J7DT 48)

Product evidence of the learner:

- representing an algorithm in program code
- creating a recursive algorithm in program code

Virtualisation Technologies (J7E0 48)

Product evidence of the learner:

- demonstrating virtualisation on a personal computer, using a cloud service to build
- deploying a virtual machine and an associated app

Network Infrastructure (J68T 48)

Product evidence of the learner installing and configuring network hosts and nodes that includes:

- client and server operating systems
- appropriate users
- groups
- permissions

Algorithms and Data Structures (J7DH 48)

Product evidence of one or more computer programs that includes the learner creating and using:

- singly and doubly linked lsts
- binary search trees and implementation of stacks
- queues
- deques
- heaps

Database Design and Development (J7DV 48)

Product evidence of a normalised relational database, appropriately populated, from a given specification along with its entity relationship diagram.

Criterion 2 descriptors

Criterion 2	Achieved	Merit	Distinction
Writes code in an industry standard programming language that implements data structures and algorithms	The learner: • implements basic algorithms (sorting, searching) and data structures (arrays, linked lists, stacks, queues) • demonstrates understanding of the operation and basic analysis of an algorithm's time and space complexity • designs and implements object-oriented programs that demonstrate understanding of classes, objects, inheritance, and polymorphism • can write code that correctly encapsulates data and behaviour	The learner: • implements a variety of algorithms and data structures, while making efficient use of resources • analyses and compares possible algorithms for a given task, referring to their space and time complexity • designs and implements well-structured, maintainable object-oriented programs that solve complex problems and demonstrate a sound grasp of object-oriented design principles and best practices • applies design patterns effectively to create code that is well-structured and reusable	The learner: • implements advanced algorithms and data structures such as trees, graphs, or hash tables in a creative and efficient manner • carries out algorithmic analysis to optimise solutions for specific constraints of space or time • designs and implements sophisticated object-oriented systems that demonstrate creativity and mastery of the paradigm • creates code that is well-structured, reusable, and demonstrates use of advanced design patterns

Criterion 2	Achieved	Merit	Distinction
Writes code in an industry standard programming language that implements data structures and algorithms (continued)	The learner: • explains the key principles of imperative and declarative programming paradigms • demonstrates a basic understanding of the differences between declarative and imperative programming paradigms and potential applications	The learner: • demonstrates a sound understanding of the fundamental principles underlying imperative and declarative programming paradigms • compares and contrasts imperative and declarative paradigms, highlighting their strengths and weaknesses	The learner: • demonstrates deep understanding of various programming paradigms (functional, logic, etc) and their use cases • critically evaluates and selects the most suitable paradigm for solving complex problems

Criterion 2 guidance

This competence relates to the professional knowledge and skills that a computer science practitioner requires when developing code that implements algorithms and standard data structures. This competence can be evidenced in the following units:

Algorithms and Data Structures (J7DH 48)

Product evidence of the learner:

- reading and writing data using different file formats for computer programs
- creating and using singly and doubly linked lists and binary search trees

- implementing stacks, queues, deques, and heap
- implementing search and sort algorithms with recursion, encryption, decryption and compression of data using existing code libraries
- implementing a time and space complexity comparison between two algorithms (either search or sort)

Object-Oriented Programming (J7DW 48)

Product evidence of the learner:

- implementing a given object-oriented design of sufficient complexity to cover the knowledge and skills for each unit outcome
- implementing at least one 'one-to-many' association, and correct use of encapsulation and inheritance
- producing test documentation that records the expected results of test data and the actual results

Programming Paradigms (J7DY 48)

Knowledge evidence of the learner:

- explaining and providing examples of imperative and declarative paradigms
- comparing low-level, procedural, object-oriented, event-driven, functional and logic languages
- writing code using an imperative programming language
- writing code using a declarative programming language

Applied Artificial Intelligence (J7DJ 48)

Knowledge evidence of learners answering time and space complexity questions.

Applied Mathematics for Computing (J7DT 48)

Product evidence of learners:

- representing an algorithm in program code
- creating a recursive algorithm in program code
- plotting a growth function using software
- performing logic operations in program code

Knowledge evidence of learners providing a solution to a growth function and Big O notation.

Criterion 3 descriptors

Criterion 3	Achieved	Merit	Distinction
Applies computational thinking and problem-solving approaches to solve real-world problems	The learner: • formulates basic problems from given scenarios and proposes simple solutions • applies problem-solving methodologies appropriately in straightforward contexts	The learner: • breaks down moderately complex problems into manageable sub-problems and creates appropriate abstractions to model real-world scenarios • adapts problem-solving strategies to suit the specific context and documents the process	The learner: • decomposes complex systems into well-defined modules, and designs high-level abstractions that effectively capture complex relationships • expertly selects and adapts problem-solving strategies and provides detailed documentation of the process

Criterion 3 guidance

This competence relates to the professional knowledge and skills that a computer science practitioner demonstrates when analysing problems and developing computing solutions. This competence can be evidenced in the following units:

Professional Practice in Computer Science (J7DX 48)

Product evidence is a project report that includes learners:

- investigating a complex computer science problem, its analysis and proposed solution
- producing a project plan containing:
  - a project timeline, project diary and recordings of project meetings
  - technical documentation relating to the solution
  - a reflection on how the project went, and their individual input in its delivery

Object-Oriented Programming (J7DW 48)

Product evidence includes learners:

- successfully investigating and applying appropriate object-oriented programming techniques
- implementing a given object-oriented design of sufficient complexity to cover the knowledge and skills for each unit outcome

- completing test documentation that records:
  - providing expected results of the test data and the actual results
  - providing evidence of using testing to amend and correct code

Applied Artificial Intelligence (J7DJ 48)

Product evidence of learners:

- interpreting at least one AI scenario or brief
- selecting and using AI services or tools

Database Design and Development (J7DV 48)

Product evidence of learners:

- analysing a problem to create an entity-relationship diagram
- performing data normalisation to third normal form
- populating a normalised relational database

Application Development for the Web (J7E1 48)

Product evidence of learners building and testing a secure full-stack web app to match a given set of client and design requirements.

Network Infrastructure (J68T 48)

Product evidence of learners:

- designing and developing a network from a given scenario
- recording network testing and troubleshooting
- providing a full network documentation
- documenting results of stringent testing that includes:
    - functionality
    - performance
    - compatibility
    - security testing
    - code that is clean and standards-compliant

Criterion 4 descriptors

Criterion 4	Achieved	Merit	Distinction
Explains the concepts and fundamental principles of an applied computing discipline; and applies appropriate tools and methodologies to design, implement, and solve specific technical problems	The learner: • explains basic concepts of an applied computing discipline and its systems • describes common use cases for an applied computing discipline, and any legal or ethical implications for their use • applies some of the technologies, tools, and methodologies in an applied computing discipline to solve a basic problem	The learner: • clearly explains the underlying concepts and fundamental principles of an applied computing discipline and its systems • clearly describes a range of use cases for an applied computing discipline, considering legal, ethical, security, and implementation implications • selects and applies appropriate technologies, tools, and methodologies to design, implement, and solve technical problems in an applied computing discipline	The learner: • comprehensively explains the concepts and techniques of an applied computing discipline and its systems • comprehensively describes a range of use cases for an applied computing discipline, and clearly sets out legal, ethical, security, and implementation implications • critically compares, selects and applies technologies, tools, and methodologies to design, implement, and efficiently solve technical problems in an applied computing discipline

Criterion 4 guidance

This competence relates to the professional knowledge and skills that a computer science practitioner requires to develop new competencies of applied computing domains. This competence can be evidenced in the following units:

Applied Artificial Intelligence (J7DJ 48)

Knowledge evidence of learners:

- explaining AI fundamentals including:
    - machine learning approaches
    - neural networks
    - algorithmic processes
    - knowledge representation
    - complexity analysis
    - model validation
    - the principles behind different AI technologies
- interpreting real-world AI scenarios
- describing AI applications across various sectors including business, healthcare, and communications
- analysing ethical implications, limitations, and societal impact of AI technologies

Product evidence of learners:

- selecting AI models and tools for specific problems
- creating training and validation of datasets
- using prompt engineering techniques to solve real-world challenges

Virtualisation Technologies (J7E0 48)

Knowledge evidence of learners:

- explaining virtualisation concepts including SaaS, PaaS, IaaS
- demonstrating an understanding of network virtualisation principles, security challenges, and the technical foundations of virtual machine management

Product evidence of learners:

- practically applying virtualisation through comparative analysis of SaaS providers and cloud deployment scenarios that covers:
    - implementation challenges
    - security implications
    - the practical benefits of virtualisation technologies in modern computing environments
    - providing virtualisation solutions using cloud platforms, deployment and management of virtual machines, and configuring virtual services

Database Design and Development (J7DV 48)

Knowledge evidence of database theory that demonstrates learners' understanding of:

- data modelling
- normalisation principles
- relational schema design
- SQL and NoSQL paradigms
- transaction processing concepts
- the theoretical foundations of database management systems

Product evidence of learners:

- creating a database solution for a real-world scenario
- implementing security measures through user management and access control
- demonstrating an understanding of:
  - data integrity
  - privacy considerations
  - the practical implications of database design decisions
  - complex queries
  - security measures
  - custom functions

Application Development for the Web (J7E1 48)

Knowledge evidence that demonstrates learners’ understanding of:

- web development concepts:
  - the client-server model
  - version control principles
  - full-stack architecture
- web technologies
- security considerations
- testing methodologies
- the technical foundations of modern web application development

Product evidence of learners:

- building a secure, responsive web application that considers:
  - accessibility
  - performance
  - security requirements
- documenting testing approaches
- building a full-stack web application using current technologies
- implementing version control workflows

Network Infrastructure (J68T 48)

Knowledge evidence that shows learner’s understanding of:

- network topology design
- IP addressing schemes
- network service architecture
- security principles, hardware functionality, and the theoretical foundations of network infrastructure including:
    - routing
    - switching
    - network management concepts

Product evidence of learners:

- designing and implementing a network solution that addresses a real-world requirement including:
    - security zones
    - service provision
    - user access management
    - hardware configuration
    - security implementation
    - service deployment

Criterion 5 descriptors

Criterion 5	Achieved	Merit	Distinction
Contributes to project management and teamwork	The learner: • creates basic project plans and implements them with guidance, using simple project management tools • participates in team activities, communicates adequately, and shows basic awareness of team dynamics • provides basic reflections on project outcomes and personal contributions, identifying obvious strengths and weaknesses	The learner: • independently develops project plans, implements them effectively, and demonstrates proficient use of various project management tools and methods • actively contributes to team success, demonstrates good interpersonal skills, communicates clearly, and shows empathy towards team members' perspectives • reflects critically on project outcomes and personal contributions, analysing strengths and areas for improvement, and proposing specific actions for future enhancement	The learner: • designs and executes comprehensive project plans, using resources efficiently, and adapts project management tools and methods to secure outcomes • takes leadership roles when appropriate, facilitates team cohesion, resolves conflicts effectively, and contributes to an inclusive environment • demonstrates critical reflection, providing insightful analysis of project outcomes, personal contributions and lessons learned and uses this to develop strategic plans for improvement

Criterion 5	Achieved	Merit	Distinction
Contributes to project management and teamwork (continued)	The learner: • uses simple collaboration tools as required	The learner: • effectively uses collaboration tools to share work and communicate with team members	The learner: • leverages advanced features of collaboration tools to enhance team productivity and communication

Criterion 5 guidance

This competence relates to the professional knowledge and skills that a computer science practitioner requires. Learners demonstrate maturity, adaptability, and accountability through the team-based project in Professional Practice in Computer Science. This competence can be evidenced in the following units:

Professional Practice in Computer Science (J7DX 48)

Product evidence of learners:

- producing a project plan containing a project timeline, project diary and recordings of project meetings
- reflecting on:
  - how the project wen
  - the challenges met and overcome
  - the ethical considerations of the team

    - their own input in the project’s delivery and how well the project went

Performance evidence of:

- your recorded observations of each individual learner's participation over the course of the project, including the value of their contributions and evidence of professional behaviours
- your evaluation of a team presentation lasting 30 minutes or more that explains the challenges met and overcome in the project, along with ethical and sustainability considerations

Object-Oriented Programming (J7DW 48)

Product evidence is completed test documentation that records both the expected results of the test data and the actual results, along with an evaluation of the outcomes from test runs.

Application Development for the Web (J7E1 48)

Product evidence is of learners using a tool for sharing, collaborating, and repositing code.

Criterion 6 descriptors

Criterion 6	Achieved	Merit	Distinction
Prepares documentation and communicates technical solutions	The learner: • prepares basic project reports and documentation that meet minimum requirements for clarity and completeness • delivers basic presentations that cover the main points to be communicated • applies formatting conventions in code, including basic comments • creates code documentation that includes simple explanations for functions and variables	The learner: • produces detailed, well-structured reports and clear documentation • delivers presentations effectively with appropriate use of technical language and visual aids • applies formatting conventions consistently in code, with clear comments to explain its purpose and functionality • creates code documentation that includes detailed explanations for functions, including parameters and return values	The learner: • creates comprehensive, professional-quality reports and documentation that exceed expectations • presents findings with exceptional clarity, engaging delivery, and insightful analysis • demonstrates exemplary code organisation and formatting that enhances readability, with insightful comments that explain both the 'how' and 'why' of the code • creates code documentation that includes thorough explanations for all functions, including edge cases and usage examples

Criterion 6 guidance

This competence relates to the professional knowledge and skills that a computer science practitioner requires when collaborating with other team members to deliver a project in computer science. This competence can be evidenced in the following units:

Professional Practice in Computer Science (J7DX 48)

Product evidence of learners producing:

- a project plan containing a project timeline, project diary and recordings of project meetings
- a comprehensive report covering all phases of the project

Performance evidence of learners:

- working as part of a team to:
    - demonstrate their final solution to a computer science problem to a client audience
    - explain challenges met and overcome in the project, along with ethical and sustainability considerations

Object-Oriented Programming (J7DW 48)

Product evidence of documented code, as part of implementing a solution from an object-oriented design.

Application Development for the Web (J7E1 48)

Product evidence of learners:

- documenting results of stringent testing that covers:
  - functionality
  - performance
  - compatibility
  - security testing
  - a completed web app that demonstrates clean, standards-compliant, and fully documented code

Network Infrastructure (J68T 48)

Product evidence of documentation for a network solution and its maintenance

Criterion 7 descriptors

Criterion 7	Achieved	Merit	Distinction
Demonstrates adherence to legal, ethical and security standards in computer science	The learner: • configures basic security settings to protect systems they use • performs basic auditing of system activities • demonstrates basic understanding and application of ethical and legal considerations in their work • implements basic encryption and data protection algorithms	The learner: • configures advanced security settings effectively, ensuring robust protection for systems they use • conducts audits and create reports on system activities • applies ethical and legal considerations effectively in a range of practical scenarios • demonstrates proficiency in using encryption and data protection algorithms	The learner: • demonstrates expertise in security configuration, providing comprehensive security documentation for systems they use • demonstrates advanced auditing skills, identifying and addressing security issues, providing comprehensive audit documentation • demonstrates deep understanding and application of ethical and legal considerations, with analysis and documentation • demonstrates advanced knowledge and implementation of encryption and data protection, with analysis and documentation

Criterion 7 guidance

This competence relates to the professional knowledge and skills that a computer science practitioner requires when working on computer science applications where there are legal, ethical or security implications. This competence can be evidenced in the following units:

Computer Science (J68M 48)

Knowledge evidence of learners evaluating at least one contemporary ethical issue in computer science.

Professional Practice in Computer Science (J7DX 48)

Product evidence of learners personally reflecting on how the project went, the challenges met and overcome, and the ethical considerations of the team.

Algorithms and Data Structures (J7DH 48)

Product evidence of learners encrypting, decrypting and compressing data using existing code libraries.

Applied Artificial Intelligence (J7DJ 48)

Knowledge evidence that demonstrates learners’ understanding of the legal, social and ethical implications of AI.

Network Infrastructure (J68T 48)

Product evidence of learners implementing the following network security features:

- dedicated devices (firewall)
- host security (firewall and/or antivirus)
- DMZ and access control
- implementing wireless technology encryption

Learners must also keep reports of network maintenance and monitoring; including records of network testing and troubleshooting.

Applied Mathematics for Computing (J7DT 48)

Knowledge evidence of learners’ application of number theory in cryptography.

Criterion 8 descriptors

Criterion 8	Achieved	Merit	Distinction
Develop meta-skills	The learner adequately engages with the process of meta-skills development in the context of the qualification by: • carrying out self-assessment of meta-skills, giving reasons for ratings or judgements made • setting clear and measurable goals, plus action strategies to develop meta-skills in all three categories • using reflective practice strategies to track progress and analyse the links between course activities, experiences and meta-skills development	The learner adequately engages with the process of meta-skills development in the context of the qualification by: • carrying out self-assessment of meta-skills, giving some insightful reasons for ratings or judgements made • setting clear and measurable goals, plus action strategies to develop meta-skills in all three categories • using reflective practice strategies to track progress and demonstrate some insight into the impact of their course activities and experiences on their meta-skills development	The learner adequately engages with the process of meta-skills development in the context of the qualification by: • carrying out self-assessment of meta-skills, giving some insightful reasons for ratings or judgements made • setting clear and measurable goals, plus action strategies to develop meta-skills in all three categories, and updating these as required • using reflective practice strategies very effectively to track progress and demonstrate insight into the impact of their course activities and experiences on their meta-skills development

Criterion 8 guidance

Maps to all units.

You must refer to the meta-skills assessment guidance when grading meta-skills. You can find meta-skills teaching, learning and assessment resources on SQA’s meta-skills web page.

Competence in individual meta-skills is not being judged here, for example the quality of a learner’s feeling or creativity. Rather, it is the process of development the learner goes through — planning, developing, and reflecting — that should be evidenced and assessed.

Although a meta-skills outcome is located in one unit, evidence of meta-skills development can be gathered from any activity at any time during the course. For meaningful reflection to take place, the process of meta-skills development should happen continually throughout the course. The range of contexts in which this can happen is very wide, and dependent on the sector, as well as individual preferences. Each unit signposts opportunities for meta-skills development.

Additional grading guidance

Grading model

The competence criteria reflect the academic, technical, and professional skills and behaviours learners should demonstrate in their performance in this qualification. The competence criteria are described in generic terms, so you can apply them regardless of which optional units a learner completes. This allows you to use evidence from any mandatory or optional unit when evaluating the competencies, as indicated in the grading matrix.

Each criterion has a grading matrix entry with performance statements at three levels in the form of a rubric that will help you evaluate and grade consistently. There is separate guidance on grading the meta-skills competence criterion in Meta-skills — assessment and grading information for centres.

When grading an individual criterion, you should refer to the grading matrix, which identifies where relevant evidence is most likely to be found across the course units. You should determine which rubric statement best reflects the quality and depth of the learner’s submitted evidence for each contributing unit. Where multiple units contribute to a single criterion (for example, ’prepares documentation and communicates technical solutions‘), you should use the highest level of performance demonstrated in any unit to inform the grade. For instance, if the evidence for Unit A indicates that ’documentation meets minimum requirements for clarity and completeness‘ while the evidence for Unit B indicates that ’documentation exceeds expectations‘, then the evidence from Unit B should be applied in forming a grade judgment for this criterion. We have provided a more complete example for the criterion below. For this example, you would assign a grade of Merit to the criterion.

Criterion 6	Achieved	Merit	Distinction
Prepares documentation and communicates technical solutions	The learner: • prepares basic project reports and documentation that meet minimum requirements for clarity and completeness • delivers basic presentations that cover the main points to be communicated • applies formatting conventions in code, including basic comments • creates code documentation that includes simple explanations for functions and variables	The learner: • produces detailed, well-structured reports and clear documentation • delivers presentations effectively with appropriate use of technical language and visual aids • applies formatting conventions consistently in code, with clear comments to explain its purpose and functionality • creates code documentation that includes detailed explanations for functions, including parameters and return values	The learner: • creates comprehensive, professional-quality reports and documentation that exceed expectations • presents findings with exceptional clarity, engaging delivery, and insightful analysis • demonstrates exemplary code organisation and formatting that enhances readability, with insightful comments that explain both the 'how' and 'why' of the code • creates code documentation that includes thorough explanations for all functions, including edge cases and usage examples

In the example above, Unit B provides evidence of performance at Distinction level in preparing reports and documentation, while Unit C provides evidence of performance at Merit level in code documentation and formatting. The contributions from Unit A are ignored as they provide evidence at a lower level that Unit B and Unit C.

The tables below illustrate the process of moving from the evidence of performance level as described in the individual rubrics to an assigned grade for each criterion. These example is for illustrative purposes only.

The process requires the exercise of your professional judgment, taking into account the relative contributions that each rubric makes to the key competence expressed in the criterion.

Criterion 1

Achieved	Merit	Distinction
Rubric 1 Achieved	**Rubric 1 Merit**	Rubric 1 Distinction
Rubric 2 Achieved	Rubric 2 Merit	Rubric 2 Distinction
Rubric 3 Achieved	**Rubric 3 Merit**	Rubric 3 Distinction

Assigned grade: Merit

Criterion 2

Achieved	Merit	Distinction
Rubric 1 Achieved	Rubric 1 Merit	**Rubric 1 Distinction**
Rubric 2 Achieved	**Rubric 2 Merit**	Rubric 2 Distinction
Rubric 3 Achieved	**Rubric 3 Merit**	Rubric 3 Distinction
Rubric 4 Achieved	Rubric 4 Merit	Rubric 4 Distinction

Assigned grade: Merit

Criterion 3

Achieved	Merit	Distinction
Rubric 1 Achieved	Rubric 1 Merit	Rubric 1 Distinction
Rubric 2 Achieved	Rubric 2 Merit	Rubric 2 Distinction
Rubric 3 Achieved	**Rubric 3 Merit**	Rubric 3 Distinction

Assigned grade: Achieved

Criterion 6

Achieved	Merit	Distinction
Rubric 1 Achieved	**Rubric 1 Merit**	Rubric 1 Distinction
Rubric 2 Achieved	Rubric 2 Merit	**Rubric 2 Distinction**
Rubric 3 Achieved	Rubric 3 Merit	**Rubric 3 Distinction**
Rubric 4 Achieved	**Rubric 4 Merit**	Rubric 4 Distinction

Assigned grade: Distinction

Criterion 7

Achieved	Merit	Distinction
Rubric 1 Achieved	Rubric 1 Merit	Rubric 1 Distinction
Rubric 2 Achieved	**Rubric 2 Merit**	Rubric 2 Distinction
Rubric 3 Achieved	Rubric 3 Merit	Rubric 3 Distinction

Assigned grade: Achieved

After completing the grading matrix for a learner using available evidence, you assign the whole qualification grade holistically. You must consider the relative contribution of each criterion to the overall aims of the qualification. The first two criteria in the Computer Science grading matrix encapsulate the learner's knowledge and understanding of computing concepts and their skills in applying them to designing and developing digital solutions. The remaining six criteria can be considered as having similar relative importance in grading.

The final grade must reflect how well the learner has demonstrated the expected academic, technical, and professional knowledge, skills and behaviours over the course of their studies, and in the work submitted as evidence.

Worked example of grading model

The table below illustrates the process of arriving at a final grade for a learner in Computer Science, adopting the holistic approach to judgment described elsewhere in this document.

Description	Achieved	Merit	Distinction
1. Demonstrates knowledge and understanding of computer systems, how they are organised and how they function.		Merit	
2. Writes code in an industry standard programming language that implements data structures and algorithms		Merit	
3. Applies computational thinking and problem-solving approaches to solve real-world problems		Merit	
4. Explains the concepts and fundamental principles of an applied computing discipline and applies appropriate tools and methodologies to design, implement, and solve specific technical problems	Achieved		
5. Contributes to Project Management and Teamwork	Achieved		

Description	Achieved	Merit	Distinction
6. Prepares documentation and communicates technical solutions		Merit	
7. Demonstrates adherence to legal, ethical and security standards in computer science			Distinction
8. Develop meta-skills		Merit	

In arriving at a final grade of MERIT for this learner, you would note that the learner does not reach Merit level in explaining concepts and fundamental principles of an applied computing discipline, nor in their contribution to project management and teamwork. However, the learner’s strengths in applying computational thinking and problem-solving to real world problems, implementing data structures and algorithms in code and demonstrating knowledge and understanding of computer systems concepts are vital competences for a computer scientist and support the decision to award a Merit grade.

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 guide.

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 email nextgen@sqa.org.uk.