Skills & Education Group Awards

In association with UK AWARDS Education Excellence

Qualification Guidance

SEG Awards Level 4 Diploma in Software Engineering

England – 610/3626/2

Qualification Guidance

About Us

At Skills and Education Group Awards we continually invest in high quality qualifications, assessments and services for our chosen sectors. As a UK leading sector specialist, we continue to support employers and skills providers to enable individuals to achieve the skills and knowledge needed to raise professional standards across our sectors.

Skills and Education Group Awards has an on-line registration system to help customers register learners on its qualifications, units and exams. In addition, it provides features to view exam results, invoices, mark sheets and other information about learners already registered.

The system is accessed via a web browser by connecting to our secure website using a username and password:

Skills and Education Group Awards Secure Login

Sources of Additional Information

Skills and Education Group Awards website www.skillsandeducationgroupawards.co.uk provides access to a wide variety of information.

Copyright

Specification Code

The specification code is D5059-04.

Issue	Date	Details of change
1.0	01 February 2024	New qualification guide

This guide should be read in conjunction with the Indicative Content document which is available on our secure website using the link above.

Skills & Education Group Awards

Qualification Guidance

Contents

This is a live document and as such will be updated when required. It is the responsibility of the approved centre to ensure the most up-to-date version of the Qualification Guide is in use. Any amendments will be published on our website and centres are encouraged to check this site regularly.

Skills & Education Group Awards

Qualification Guidance

Introduction

The SEG Level 4 Diploma in Software Engineering covers the method of developing software. Learners will learn to plan, design, program, manage and test software applications and develop an understanding of the software design process. This will support a learners development throughout their career. The aim of any programme of study in software engineering should be to develop expertise in software development.

The key areas covered include:

- Computer Architecture and Mathematics for Software Engineers
- Web Programming
- Software Engineering Concepts and Principles
- Object Oriented Programming Concepts
- Database Management Systems
- Software Project

The knowledge and skills gained will prepare learners to progress onto higher programmes of study, and related qualifications, in Computing and Information Technology.

Pre-requisites

There are no entry requirements for this qualification. However, learners should be working to at **least** Level 3.

Skills and Education Group Awards expects approved centres to recruit with integrity on the basis of a trainee's ability to contribute to and successfully complete all the requirements of a unit(s) or the full qualification.

Skills & Education Group Awards

Qualification Guidance

Qualification Structure and Rules of Combination

Rules of Combination: Level 4 Diploma in Software Engineering

Learners must achieve **all** 120 credits from **all** the 6 mandatory units.

Unit	Unit Number	Level	Credit Value	GL
Mandatory Group Min Credit Target - 120				
Software Engineering and Databases	J/650/9605	4	20	80
Productivity and Networks	K/650/9606	4	20	80
IT Concepts and Hardware	L/650/9607	4	20	80
Foundations of Web Design	M/650/9608	4	20	80
Foundations of Programming	R/650/9609	4	20	80
Final Project	A/650/9610	4	20	80

Aims

Upon successful completion of the SEG Awards Level 4 Diploma in Software Engineering, learners will be able to:

- Develop knowledge of some of the technologies and processes used in the modern software development
- Demonstrate an awareness of modern software development methodologies and their role in software development
- Analyse client requirements to develop and test software solutions based on client requirements
- Program object-oriented applications for a web-based system which uses a database system
- Work effectively in a team environment
- Communicate effectively in written and oral form

Skills & Education Group Awards

Qualification Guidance

Target Group

The SEG Awards Level 4 Diploma in Software Engineering is designed for learners, **16 years of age and over**, who are looking to develop competence as a software developer. It is designed to provide an education that will develop knowledge and understanding of relevant theories and principles together with technical skills and capabilities associated with the practice of the discipline.

Assessment

The curriculum is set up to support a portfolio approach to continuous assessment. Learners will study modules and develop a portfolio of evidence. Each module will have milestones where formative assessment is provided, and learners can then continue to work on their portfolios before a final submission at the end of the module.

For each module, an assessment grid is provided indicating the learning outcomes to be achieved and the evidence required to support their attainment. This grid contains evidence requirements for grading at pass, merit, and distinction. The criteria are cumulative, so to achieve a merit grade a learner must satisfy the criteria for both a pass and for a merit. Similarly, to achieve a distinction grade a learner must satisfy, pass, merit, and distinction criteria.

To achieve a pass in a module, a pass grade must be attained for all learning outcomes. The overall grade for each module will be determined by the predominant attainment in each of the learning outcomes. For example, most modules have four learning outcomes so if three are attained at merit, then a merit grade is the outcome. If the outcome is that two learning outcomes are graded pass and two at merit, then a merit for the module would be awarded. For a distinction grade, the predominant attainment in each of the learning outcomes must be at distinction grade with all learning outcomes achieving at least a merit grade.

For the diploma to be awarded, a pass grade must be achieved in all modules. The overall grade for the diploma will be determined based on the predominant outcome for each of the modules. There are six modules, so to achieve an overall grade of merit at least three modules must be graded at merit. To achieve a distinction, all modules must be graded at minimum of merit and at least three at distinction.

Practice Assessment Material

Skills and Education Group Awards confirm that there is no practice material available for the SEG Awards Level 4 Diploma in Software Engineering.

Teaching Strategies and Learning Activities

The fundamental philosophy that guides this curriculum is project based learning with a balance between the following elements.

- Lectures and lessons – where knowledge is acquired
- Seminars and tutorials – where knowledge is consolidated and know-how developed
- Laboratories – where practical skills are demonstrated and developed
- Projects – where learners can develop their skills of synthesis

Centres should adopt a delivery approach which supports the development of all individuals. The aims and aspirations of all the learners, including those with identified special needs or learning difficulties/disabilities, should be considered and appropriate support mechanisms put in place.

Progression Opportunities

Learners who achieve this qualification could progress onto further Level 4 and Level 5 qualifications in IT, Computing and Software. Learners could also progress into employment.

Centres should be aware that Reasonable Adjustments which may be permitted for assessment may in some instances limit a learner's progression into the sector. Centres **must**, therefore, inform learners of any limits their learning / physical difficulty may impose on future progression.

Tutor / Assessor Requirements

Skills and Education Group Awards require those involved in the teaching and assessment process to be suitably qualified. Assessors should also be trained and qualified to assess or be working towards appropriate teaching qualifications.

Qualification Guidance

Minimum requirements when delivering this qualification: Skills and Education Group Awards expects that staff will be appropriately qualified to assess learners against the outcomes and criteria within the units. Teaching staff **must** be qualified at least a level above in a relevant subject to which they are teaching.

Those responsible for Internal Quality Assurance (IQA) **must** be knowledgeable and or qualified of the subject/occupational area to a suitable level to carry out accurate quality assurance practices and processes.

Language

This specification and associated assessment materials are in English only.

Skills & Education Group Awards

Qualification Guidance

Qualification Summary

Qualification									
SEG Awards Level 4 Diploma in Software Engineering									
Qualification Purpose	Prepare for further learning or training and/or develop knowledge and/or skills in a subject area								
Age Range	**Pre 16**		**16-18**	✓	**18+**	✓	**19+**	✓	
Regulation	The above qualification is regulated by: • Ofqual								
Assessment	• Portfolio of Evidence								
Type of Funding Available	See FaLA (Find a Learning Aim)								
Grading	Pass/Merit/Distinction/Fail								
Operational Start Date	01/02/2024								
Review Date	01/02/2027								
Operational End Date	-								
Certification End Date	-								
Guided Learning (GL)	480 hours								
Total Qualification Time (TQT)	1200 Hours								
Credit Value	120								
Skills and Education Group Awards Sector	Computing and Software								
Regulator Sector	6.1 Digital technology (practitioners)								
Support from Trade Associations	-								

Skills & Education Group Awards

Qualification Guidance

Unit Details

Software Engineering and Databases

Unit Reference	**J/650/9605**
Level	**4**
Credit Value	**20**
Guided Learning (GL)	**80 hours**
Unit Summary	This unit aims to explore and develop established concepts and methods used to model software applications. Before the application can be developed learners must understand the data that it will create and manipulate. In this unit learners will learn to design software and database solutions using current industry techniques.
Unit Aim	The learner should develop a portfolio based on a company or organisation in their country or a case study identified by their tutors.

Learning Outcomes (1 to 4)	**Assessment Criteria (1.1 to 4.3)**	
The learner will	***The learner can***	
1. Be able to design databases for business problems using appropriate notations and theories	1.1	Create well-structured databases for solving business problems to include the following design theories: a) normalisation b) indexing c) data modelling
	1.2	Use appropriate notations including Entity-Relationship Diagrams (ERD) or Unified Modelling Language (UML) to accurately represent the structure of databases, demonstrating a clear understanding of theoretical foundations and their practical application
	1.3	Evaluate the adaptability of the developed database architecture relating to changes and scalability

Qualification Guidance

	1.4	Use suitable documentation tools to communicate database designs, articulating the theoretical foundations behind design decisions for practical business needs
2. Be able to design, implement and manipulate databases using structured query language	2.1	Design databases using Structured Query Language (SQL), ensuring that the design aligns with specified requirements, including: a) table creation b) relationships c) constraints
	2.2	Create SQL queries to perform the following data related operations: a) updates b) insertions c) deletions d) joins e) subqueries
	2.3	Evaluate the efficiency of the queries developed in 2.2 in retrieving and manipulating data, relating to query performance through indexing and other optimisation techniques
	2.4	Explain how the integrity and confidentiality of the database can be assured
3. Understand the key concepts in Software Engineering	3.1	Analyse the fundamental concepts in Software Development Life Cycle (SDLC), including: a) requirements analysis b) planning c) system design d) implementation and coding requirements e) testing and integration f) deployment g) maintenance
	3.2	Explain the following key methodologies and best practices, highlighting their distinctive features and application in different

Qualification Guidance

		development contexts: a) agile development b) waterfall development
	3.3	Explain the importance of the iterative approach to software development
4. Be able to design a software application using appropriate Software Engineering theories and techniques	4.1	Apply relevant Software Engineering theories to design a software application, demonstrating an understanding of theoretical principles in areas such as software architecture, design patterns, and system modelling
	4.2	Use the following methods in the software application designed in 4.1 including: a) requirements analysis b) modularisation c) user interface designed to create a well-structured and functional system
	4.3	Justify design choices considering the following factors: a) scalability b) maintainability c) user experience

Skills & Education Group Awards

Qualification Guidance

Productivity and Networks

Unit Reference	**K/650/9606**	
Level	**4**	
Credit Value	**20**	
Guided Learning (GL)	**80 hours**	
Unit Summary	This unit aims to develop fundamental understanding of the way computer networks operate, the core components of networked computer system and as well as how network communication is structured. The unit will enable learners to develop their understanding of productivity applications.	
Unit Aim	The learner should develop a portfolio based on their learning.	
Learning Outcomes (1 to 4)	**Assessment Criteria (1.1 to 4.4)**	
The learner will	***The learner can***	
1. Understand the underlying concepts, and principles, relevant to productivity applications	1.1	Explain the foundational concepts underlying productivity applications including: a) document management b) collaboration c) data integration
	1.2	Explain the principles governing the functionality of productivity applications, highlighting how features contribute to user productivity and efficiency
	1.3	Relate the concepts and principles explained in 1.1 and 1.2 to practical applications, providing examples of how they manifest in widely used productivity tools
2. Understand the core architecture of contemporary computers and devices, associated mathematical principles, and relevant machine and data processes	2.1	Analyse the core components of contemporary computer architecture, including the: a) central processing unit (CPU) b) memory hierarchy c) storage devices d) input/output systems

Qualification Guidance

	2.2	Explain how the following mathematical principles underpin the operation of components within computer architecture: a) Boolean algebra b) binary arithmetic
	2.3	Explain the processes of instruction execution cycles, making clear how data is processed within the context of contemporary computer systems
	2.4	Relate the core architecture concepts in 2.1 to real-world computing devices, providing examples of how these principles manifest in the design and functionality of contemporary computers and devices
3. Understand the need for network protocols, services and communication models	3.1	Evaluate the importance of network protocols in facilitating communication between devices, emphasising their role in standardising data exchange and ensuring interoperability
	3.2	Explain the following essential network services clarifying how these services contribute to the reliability and efficiency of network communication: a) routing b) addressing c) error detection
	3.3	Explain the Open System Interconnection (OSI) and Transmission Control Protocol/Internet Protocol (TCP/IP) communication models and their significance in conceptualising and organising the processes involved in network communication
	3.4	Analyse two scenarios where effective network communication is crucial, and how the need for protocols, services, and

Qualification Guidance

		communication models arises in real-world contexts
4. Be able to implement fundamentals of switching and routing, including network addressing	4.1	Explain the fundamental concepts of switching and routing, including how switches forward data within local networks and routers facilitate communication between different networks
	4.2	Describe the principles of network addressing, including the use of IP addresses, subnetting, and the role of addressing in routing data between devices in a network
	4.3	Implement basic switching functionality, demonstrating the ability to configure and manage switches to enable efficient data forwarding within a local network
	4.4	Configure routing and addressing in practical scenarios, showcasing the ability to: a) set up routers b) assign IP addresses c) establish communication paths between devices in different network segments

Skills & Education Group Awards

Qualification Guidance

IT Concepts and Hardware

Unit Reference	**L/650/9607**
Level	**4**
Credit Value	**20**
Guided Learning (GL)	**80 hours**
Unit Summary	This module introduces the fundamental principles and features that underpin all modern computing systems. Learners will learn about computer system and their functions. The students will understand the core components of computer systems and how a PC is built. You will develop an awareness of their functionality, advantages, disadvantages, and relevance in different contexts.
Unit Aim	The learner should develop a portfolio based on their learning.
Learning Outcomes (1 to 4)	**Assessment Criteria (1.1 to 4.4)**
The learner will	***The learner can***
1. Understand the common elements of typical computer hardware and software upon which applications are constructed	1.1 Explain the key elements of computer hardware, including: a) central processing units (CPUs) b) memory c) storage devices d) input/output devices 1.2 Explain the various software components involved in computing and their roles in the application development process, including: a) operating systems b) middleware c) application software 1.3 Describe the interaction between computer hardware and software components, enabling hardware resources to execute applications effectively 1.4 Analyse the common elements of computer hardware and software in the context of application

		construction, demonstrating an understanding of how these components collectively contribute to the development and functionality of applications
2. Understand the core architecture of contemporary computers and devices	2.1	Explain the core components of contemporary computer architecture, including the a) central processing unit (CPU) b) memory hierarchy c) storage devices d) input/output devices
	2.2	Examine the interactions between different components within the computer architecture, illustrating how the CPU, memory, and storage work together to execute instructions and store data
	2.3	Explain the role of machine processes within the architecture, detailing how instructions are fetched, decoded, and executed, and how data is processed within the system
	2.4	Relate the core architecture concepts described in 2.1 to modern computing devices, providing examples of how these principles are used in the design and functionality of contemporary computers, laptops, tablets, and smartphones
3. Be able to apply the fundamentals of digital logic used in computational paradigms, operating systems, processing, I/O devices and memory	3.1	Use basic digital logic principles to design and implement circuits for computing tasks
	3.2	Apply simple digital logic concepts in operating systems to manage computer resources and processes
	3.3	Implement basic digital logic in processors to make them work efficiently
	3.4	Apply basic digital logic to design interfaces for devices and memory

Qualification Guidance

		systems, making data transfer and storage work effectively
4. Understand mathematical principles related to computing, and relevant machine and data processes	4.1	Explain how maths is used in computing, including how it's applied in machines and data processes
	4.2	Explain how machines in computing operate, including how instructions are carried out and how data is manipulated
	4.3	Show how data is processed in computing, including how information is input, changed, stored, and output
	4.4	Relate the mathematical concepts to practical machine and data processes, providing examples of how these ideas are used in real-world computing

Skills & Education Group Awards

Qualification Guidance

Foundations of Web Design

Unit Reference	**M/650/9608**	
Level	**4**	
Credit Value	**20**	
Guided Learning (GL)	**80 hours**	
Unit Summary	This unit is an introduction to the key concepts in web design. Learners will learn how HTML is used to structure documents and how CSS is used to style the document. Learners will follow the design process from conceptualisation and wireframing through to the coding of mark-up and styling. Learners will learn how and why JavaScript can be applied client side to add interactivity to web pages and how to use popular client-side frameworks such as jQuery.	
Unit Aim	The learner should develop their websites based on an industry case study identified by their tutors.	
Learning Outcomes (1 to 4)	**Assessment Criteria (1.1 to 4.4)**	
The learner will	***The learner can***	
1. Be able to design web User Interface (UI) and User Experience (UX) with the latest standards in HTML, CSS and JavaScript	1.1	Create a web interface that is easy for users to navigate and interact with, using the latest HTML, CSS, and JavaScript standards
	1.2	Design a visually appealing user interface (UI) that follows the latest standards to enhance the overall look and feel of the website, ensuring the design aligns with contemporary web development standards
	1.3	Use the latest HTML, CSS, and JavaScript practices for responsiveness and interactivity to ensure that users have a smooth and enjoyable user experience (UX)while interacting with the website

Qualification Guidance

2. Understand the core concepts in designing a web page using HTML	2.1	Explain of the basic elements and structure of HTML used to create a web page
	2.2	Examine how HTML tags are used to define and organise content on a web page, including: a) headings b) paragraphs c) lists d) links
	2.3	Illustrate how HTML is used to structure a web page, showing the arrangement of headers, main content, and footers
	2.4	Describe how HTML is used to integrate multimedia elements, such as images and videos, into a web page for a more engaging user experience
3. Understand how Cascade Style Sheets (CSS) can be used for consistent look and feel	3.1	Explain the basics of CSS and how it is used to style and format elements on a web page
	3.2	Illustrate how CSS properties are applied to control the appearance of elements to maintain a consistent look and feel, including: a) colours b) fonts c) spacing
	3.3	Show how CSS selectors are used to target specific elements on a web page, ensuring that styling is applied consistently across those elements
	3.4	Explain how CSS can be used for responsive design, ensuring that the look and feel of the web page adapt effectively to different screen sizes and devices
4. Be able to apply techniques to create web content that can be accessed	4.1	Describe techniques for responsive design, including how web content can adapt to different

Qualification Guidance

from various devices including mobile phones		devices to provide a seamless user experience
	4.2	Illustrate how to create mobile-friendly layouts, using techniques that ensure web content looks and functions well on smaller screens
	4.3	Apply media queries in CSS to tailor styling and layout based on the characteristics of different devices, ensuring optimal content presentation across various screens
	4.4	Use techniques to ensure cross-browser compatibility, making web content accessible from various devices and browsers

Skills & Education Group Awards

Qualification Guidance

Foundations of Programming

Unit Reference	**R/650/9609**
Level	**4**
Credit Value	**20**
Guided Learning (GL)	**80 hours**
Unit Summary	This unit introduces the learner to the computer programming. The module provides a foundation and understanding of key programming concepts and how these can be applied in a programming language. The focus is on the design and implementation of programs using the facilities of a programming language which is widely used for developing modern applications.
Unit Aim	The learner should develop their programs based on a case study identified by their tutors.
Learning Outcomes (1 to 4)	**Assessment Criteria (1.1 to 4.4)**
The learner will	***The learner can***
1. Understand the key concepts that relate to designing and implementing small programs within the procedural programming paradigm	1.1 Explain the basics of procedural programming, including key concepts used in designing and implementing small programs 1.2 Illustrate how a program is structured in the procedural programming paradigm, showing the arrangement of procedures, functions, or methods to accomplish tasks 1.3 Describe how variables are used within procedural programming, including how data is stored and manipulated to achieve specific goals 1.4 Explain control flow concepts, showing how decisions (if statements) and loops are used to control the execution of instructions in a small program

Qualification Guidance

2. Be able to use essential features of a mainstream procedural programming language (Python) to implement solutions to a variety of programming problems	2.1	Explain the following key features of the Python programming language including how they work: a) variables c) data types d) functions
	2.2	Illustrate how programs are structured in Python, showing the arrangement of code blocks, functions, and statements in the procedural programming style
	2.3	Use variables and various data types in Python to implement solutions to programming problems, including the effective and efficient manipulation and storage of data
	2.4	Implement solutions to two programming problems using functions in Python, which includes designing modular and reusable code
3. Be able to apply appropriate Software Development Life Cycle (SDLC) and program testing techniques on small programs	3.1	Create a small program following the steps of the Software Development Life Cycle (SDLC) from planning to implementation
	3.2	Apply testing methods to ensure small programs function as intended, highlighting how to identify and fix errors or bugs through testing
	3.3	Use version control techniques to track changes and manage the development process of small programs, ensuring a systematic approach to modifications
	3.4	Perform debugging strategies to identify and rectify errors effectively when encountering issues in small programs

Qualification Guidance

4. Be able to use appropriate software tools and program development environments	4.1	Justify the selection of software tools that are appropriate for a given task
	4.2	Use integrated program development environments (IDEs) for writing, testing, and debugging code
	4.3	Use specialised code editors for writing and managing code
	4.4	Explain the importance of using software tools and development environments proficiently, to ensure a smooth and efficient workflow in program development

Skills & Education Group Awards

Qualification Guidance

Final Project

Unit Reference	**A/650/9610**
Level	**4**
Credit Value	**20**
Guided Learning (GL)	**80 hours**
Unit Summary	This unit is intended as a foundation study of the theories of organisation and management to the extent that they apply to software projects. Learners will examine the theories of management as they apply to various types of organisation within a software engineering context and develop an understanding of how these impact on the software application development. Learners will develop an understating of the function and management of teams and leadership through role play.
Unit Aim	The learner should develop a portfolio based on a company or organisation in their country or a case study identified by their tutors.
Learning Outcomes (1 to 4)	**Assessment Criteria (1.1 to 4.3)**
The learner will	***The learner can***
1. Understand how a software project is planned and executed including the primary roles and functions in the process	1.1 Explain the basics of how software projects are planned, including the initial steps involved in organising and outlining project goals 1.2 Describe the steps involved in executing a software project, including how tasks are carried out to meet project objectives 1.3 Explain the primary roles involved in a software project, including the different responsibilities individuals undertake 1.4 Explain the main functions performed in the software project process, including: a) planning b) requirement analysis

		c) designing e) coding f) testing g) documentation h) deployment i) Maintenance
2. Be able to analyse problem specifications and design appropriate solutions	2.1 2.2 2.3 2.4	Identify the key requirements and constraints outlined in a given scenario Analyse different elements of a problem, breaking it down into manageable parts to comprehend its complexity and revealing potential solutions Design appropriate solutions based on the analysis of the problem in 2.2, using effective strategies to address identified issues Evaluate the effectiveness of the solutions and strategies designed and used in 2.3
3. Understand the theories of teamwork and communication in software projects and explain the factors that affect the effective functioning of teams	3.1 3.2	Explain the theories related to teamwork and communication in software projects highlighting strategies that enhance effective information exchange within teams Examine factors that can influence the effective functioning of teams in software projects, including potential challenges and opportunities for improvement
4. Be able to apply software development techniques and use suitable technologies to create prototype solutions to a variety of problems	4.1 4.2	Apply basic software development techniques to create prototype solutions, including coding, testing, and debugging principles, to two given problems Use appropriate technologies for creating prototype solutions, selecting tools and frameworks that match the requirements of the given problems

Qualification Guidance

	4.3	Apply software development techniques and technologies to tailor prototype solutions to specific problem scenarios, demonstrating adaptability and problem-solving skills

Skills & Education Group Awards

Qualification Guidance

Recognition of Prior Learning (RPL), Exemptions, Credit Transfers and Equivalencies

Skills and Education Group Awards policy enables learners to avoid duplication of learning and assessment in a number of ways:

- Recognition of Prior Learning (RPL) – a method of assessment that considers whether a learner can demonstrate that they can meet the assessment requirements for a unit through knowledge, understanding or skills they already possess and do not need to develop through a course of learning.
- Exemption - Exemption applies to any certificated achievement which is deemed to be of equivalent value to a unit within Skills and Education Group Awards qualification but which does not necessarily share the exact learning outcomes and assessment criteria. It is the assessor's responsibility, in conjunction with the Internal Moderator, to map this previous achievement against the assessment requirements of the Skills and Education Group Awards qualification to be achieved in order to determine its equivalence. Any queries about the relevance of any certificated evidence, should be referred in the first instance to your centre's internal moderator and then to Skills and Education Group Awards.
  It is important to note that there may be restrictions upon a learner's ability to claim exemption or credit transfer which will be dependent upon the currency of the unit/qualification and a learner's existing levels of skill or knowledge.
  Where past certification only provides evidence that could be considered for exemption of part of a unit, learners must be able to offer additional evidence of previous or recent learning to supplement their evidence of achievement.
- Credit Transfer – Skills and Education Group Awards may attach credit to a qualification, a unit or a component. Credit transfer is the process of using certificated credits achieved in one qualification and transferring that achievement as a valid contribution to the award of another qualification. Units/Components transferred must share the same learning outcomes and assessment criteria along with the same unit number. Assessors must ensure that they review and verify the evidence through sight of:
  - Original certificates OR
  - Copies of certificates that have been signed and dated by the internal moderator confirming the photocopy is a real copy and make these available for scrutiny by the External Moderator.
- Equivalencies – opportunities to count credits from the unit(s) from other qualifications or from unit(s) submitted by other recognised organisations towards the place of mandatory or optional unit(s) specified in the rule of combination. The unit must have the same credit value or greater than the unit(s) in question and be at the same level or higher.

Skills and Education Group Awards encourages its centres to recognise the previous achievements of learners through Recognition of Prior Learning (RPL),

Exemption, Credit Transfer and Equivalencies. Prior achievements may have resulted from past or present employment, previous study or voluntary activities. Centres should provide advice and guidance to the learner on what is appropriate evidence and present that evidence to the external moderator in the usual way.

Further guidance can be found in 'Delivering and Assessing Skills and Education Group Awards Qualifications' which can be downloaded from https://skillsandeducationgroupawards.co.uk/for-centres/

Certification

Learners will be certificated for all units and qualifications that are achieved and claimed.

Skills and Education Group Awards' policies and procedures are available on the website.

Exemptions

This qualification contains no exemptions. For further details see Recognition of Prior Learning (RPL), Exemptions, Credit Transfers and Equivalencies.

Glossary of Terms

GL (Guided Learning)
GL is where the learner participates in education or training under the immediate guidance or supervision of a tutor (or other appropriate provider of education or training). It may be helpful to think – 'Would I need to plan for a member of staff to be present to give guidance or supervision?'

GL is calculated at qualification level and not unit/component level.

Examples of Guided Learning include:

- Face-to-face meeting with a tutor
- Telephone conversation with a tutor
- Instant messaging with a tutor
- Taking part in a live webinar
- Classroom-based instruction
- Supervised work
- Taking part in a supervised or invigilated formative assessment
- The learner is being observed as part of a formative assessment.

TQT (Total Qualification Time)
'The number of notional hours which represents an estimate of the total amount of time that could reasonably be expected to be required, in order for a learner to achieve and demonstrate the achievement of the level of attainment necessary for the award of a qualification.' The size of a qualification is determined by the TQT.

TQT is made up of the Guided Learning (GL) plus all other time taken in preparation, study or any other form of participation in education or training but not under the direct supervision of a lecturer, supervisor or tutor.

TQT is calculated at qualification level and not unit/component level.

Examples of unsupervised activities that could contribute to TQT include:

- Researching a topic and writing a report
- Watching an instructional online video at home/e-learning
- Watching a recorded webinar
- Compiling a portfolio in preparation for assessment
- Completing an unsupervised practical activity or work
- Rehearsing a presentation away from the classroom
- Practising skills unsupervised
- Requesting guidance via email – will not guarantee an immediate response.