T-LEVELS | Institute for Apprenticeships & Technical Education

T LEVEL

Technical Qualification in Digital Production, Design and Development

Specification

First teaching from September 2020
Version 1.1 January 2022

Pearson

T Level Technical Qualification in Digital Production, Design and Development

Specification

First teaching September 2020

Version 1.1 January 2022

About Pearson

Pearson is the world's leading learning company, with 25,000 employees in more than 70 countries working to help people of all ages to make measurable progress in their lives through learning. We put the learner at the centre of everything we do, because wherever learning flourishes, so do people. Find out more about how we can help you and your learners at qualifications.pearson.com

This specification is version 1.1. We will inform providers of any changes to this version through bulletins and provider updates. The latest version can be found on our website.

ISBN 978 1 446 96851 2

Welcome to career-ready education

T Levels are new two-year, Level 3 study programmes that will follow the study of GCSEs and Technical Awards at Key Stage 4 and offer an attractive alternative to A Levels and Apprenticeships. T Levels will combine classroom theory, practical learning and a minimum 315 hours of industry placement with an employer to make sure students have real experience of the workplace.

The Technical Qualification (TQ) is the main classroom-based element of the T Level and will see students learning from a curriculum that has been shaped by industry experts. During the two-year programme, students will learn the core knowledge that underpins each industry and they will also develop occupationally specific skills that will allow them to enter skilled employment within a specific occupation.

The T Level programmes have been developed in collaboration with employers and businesses so the content will meet the needs of industry and prepare students for work. They provide the knowledge and experience needed to open the door to highly skilled employment, an Apprenticeship or higher level study, including university.

Technical Qualification and collaboration

The Outline Content for the *T Level Technical Qualification in Digital Production, Design and Development* has been produced by T Level panels of employers, professional bodies and Providers, based on the same standards as those used for Apprenticeships. Employers involved in designing the Outline Content include Fujitsu, Accenture, Lloyds Banking Group, Geo Strategies, CGI, Accordio, Strategic Discourse Ltd. and ODAG Consultants.

Pearson has used the Outline Content to form the basis of the Technical Qualification specification. This includes:

- elaboration of the Outline Content to provide a specification that gives Providers an accurate interpretation of what is required to be taught and assessed
- enabling students to achieve threshold competence in relation to each Occupational Specialist Component
- the integration of English, maths and digital content.

Students who complete a *T Level Technical Qualification in Digital Production, Design and Development* will be able to choose between moving into a skilled occupation or further study; for example, a higher or degree level Apprenticeship, or higher-level technical study, including higher education. Therefore it was essential we developed the qualification in close collaboration with experts from professional bodies, businesses and universities, and with the Providers who will be delivering the qualification.

Our engagement with experts during the development of the qualification ensures the content will meet your needs and give students quality preparation to help them progress. We are grateful to all university and further-education lecturers, teachers, employers, professional body representatives and other individuals who have generously shared their time and expertise to help us develop these new qualifications.

Employers, professional bodies and higher-education providers who have worked with us include:

- ARM
- BBC
- BT
- Cisco
- CompTIA
- Nationwide
- Siemens.

Summary of changes	
Change to title to show version publication date, for clarity.	Front cover
Insertion of specification version.	Title page
Removal of Reference to third party material statement, for clarity.	Imprint page
Removal of statement 'students will need to achieve a minimum standard of English and Maths, either GCSE grade 4 or above, or level 2 Functional skills', this is no longer a requirement of the T level.	Throughout
Addition of Unclassified grade, for clarity.	Throughout
Addition of Transferring between T levels statement, for information.	Page 7
Minor formatting changes to the main body and footer of the specification.	Throughout

Contents

1. Qualification summary and key information

T Level Technical Qualification in Digital Production, Design and Development							
Total Guided Learning Hours (GLH)		1200 GLH					
Total Qualification Time (TQT)		1640 TQT					
First registration:	September 2020	**Recommended age range:**			16–19		
Core Component:	600 GLH 810 TQT	**Grade:**	A*–E and Unclassified				
Assessment component	**Assessment method**	**Duration**	**Marks**	**Weighting**	**Timetable**	**Availability**	**Marking approach**
Core Paper 1: Digital Analysis, Legislation and Emerging Issues	Written examination paper	2.5 hours	100	33.33%	Set date and time	May/June November	Externally marked
Core Paper 2: The Business Environment	Written examination paper	2.5 hours	100	33.33%	Set date and time	May/June November	Externally marked
Employer Set Project	Externally set project	14.5 hours	100	33.33%	Window	May/June November	Externally marked
Occupational Specialist Component:	600 GLH 830 TQT	**Grade:**	P, M, D and Unclassified				
Digital Production, Design and Development	Externally set project	67 hours	145	100%	Task-specific: window/set date and time	February/May	Externally marked

If a student completes the assessments but is not successful in reaching the minimum threshold for the Core and Occupational specialism component, they will be issued with a U grade.

T Level Technical Qualification in Digital Production, Design and Development			
Qualification Number (QN)	603/5832/4		
First Registration Date	September 2020		
Approved Age Range	16–19		
Total Guided Learning Hours (GLH)	1200 GLH*		
Total Qualification Time (TQT)	1640 TQT*		
Assessment	All assessments are externally set and marked by Pearson		
Grading Overview	**Core**	**Occupational specialism**	**Overall**
	All grades for this component will be on a scale of A*–E and Unclassified	All grades for this component will be on a scale of Pass, Merit, Distinction and Unclassified	The overall grade will be on a scale of Unclassified, Pass, Merit, Distinction, Distinction* #

* See Section 2 below for further information about GLH and TQT.

Pearson will not award the overall grade for the Technical Qualification. The overall grade will be awarded by the Institute for Apprenticeships and Technical Education (the Institute). See *Section 6 Technical Qualification grading*, *T Level grading and results reporting* for further information. If a student completes the assessments but is not successful in reaching the minimum threshold for the core and/or occupational specialism component, they will be issued with a U grade.

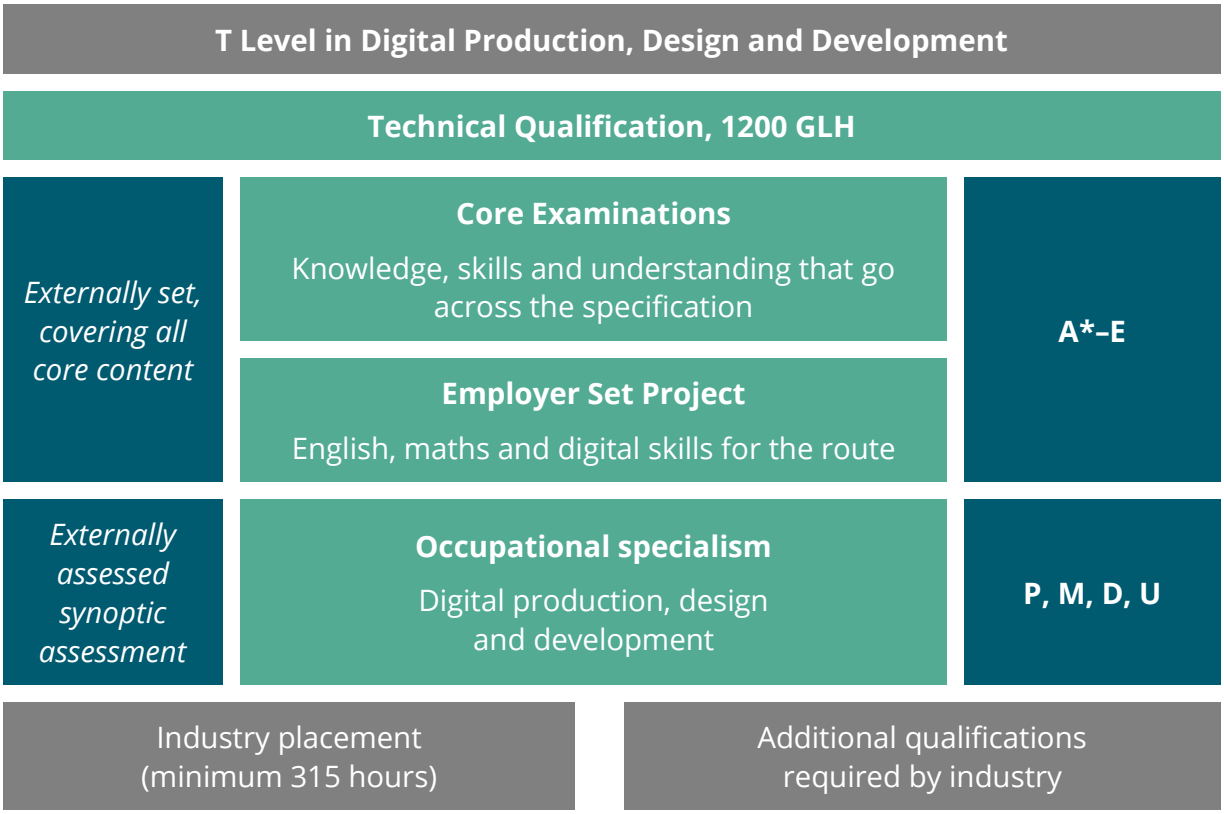
2. Introduction to the T Level Technical Qualification in Digital Production, Design and Development

This specification contains all the information you need to deliver the *T Level Technical Qualification in Digital Production, Design and Development*.

This qualification forms the substantive part of the *T Level in Digital Production, Design and Development*, which includes other elements that are required to be successfully completed in order for students to be awarded the T Level from the Institute for Apprenticeships and Technical Education (the Institute).

The *T Level in Digital Production, Design and Development* will include:

- a 315-hour industry placement that is related to the digital sector.



Qualification structure

The *T Level Technical Qualification in Digital Production, Design and Development* has two mandatory components:

1. Core Component

This component covers the underpinning knowledge, concepts and skills that support threshold competence in the digital industry. It has 600 GLH and is assessed by two externally set Core examinations and an Employer Set Project.

The content and details of each of these assessments is provided in *Section 4 Core Component*.

2. Occupational Specialist Component

There is a single Occupational Specialist Component in this Technical Qualification, which is 600 GLH. Therefore, students undertaking the *T Level Technical Qualification in Digital Production, Design and Development* will choose this specialism.

This component covers the occupational specialist knowledge and skills required to demonstrate threshold competence for the specialism and it will be assessed by a skills-related project that synoptically assesses the Performance Outcome skills and associated underpinning knowledge.

The content and details of the assessment for the Occupational Specialist Component is provided in Section 5.

Total Qualification Time (TQT) and Guided Learning Hours (GLH)

For all regulated qualifications, we specify a total number of hours that students are expected to undertake in order to complete and show achievement for the qualification – this is the Total Qualification Time (TQT). The TQT value indicates the size of a qualification.

Within the TQT, we identify the number of Guided Learning Hours (GLH) that a centre delivering the qualification needs to provide. Guided learning means activities that directly or immediately involve tutors and assessors in teaching, supervising and invigilating students, for example lectures, tutorials, online instruction and supervised study.

As well as guided learning, there may be other required learning that is directed by tutors or assessors. This includes, for example, private study, preparation for assessment and undertaking assessment when not under supervision, such as preparatory reading, revision and independent research. TQT and GLH are assigned after consultation with users of the qualifications.

The TQT and GLH for this qualification and the two components are shown below:

TQT:

- The *T Level Technical Qualification in Digital Production, Design and Development* has a TQT value of 1640.
- The Core Component has a TQT value of 810.
- The Occupational Specialist Component has a TQT value of 830.

GLH:

- The *T Level Technical Qualification in Digital Production, Design and Development* has a GLH value of 1200.
- The Core Component has a GLH value of 600.
- The Occupational Specialist Component has a GLH value of 600.

T Level Technical Qualification in Digital Production, Design and Development			
Total Guided Learning Hours (GLH)		**Total Qualification Time (TQT)**	
1200 GLH		1640 TQT	
Core Component		**Occupational Specialist Component**	
GLH	**TQT**	**GLH**	**TQT**
600 GLH	810 TQT	600 GLH	830 TQT

Technical Qualification aims and purpose

This Technical Qualification is for T Level students who are undertaking the *T Level in Digital Production, Design and Development*. It is intended for students who want to progress to a career in the Digital sector, with a focus on software design and development.

The purpose of the *T Level Technical Qualification in Digital Production, Design and Development* is to ensure students have the knowledge and skills needed to progress into highly skilled employment, an Apprenticeship or higher level study, including university, within the specialist area of software design and development.

At the end of the Technical Qualification, students are expected to demonstrate threshold competence, which means that they have gained the core knowledge and skills related to software design and development and are well placed to develop full occupational competence with additional development and support once in employment in the digital sector.

Student profile and progression

Students undertaking this Technical Qualification will be 16–19 years old and in full-time education. They will have chosen a T Level as an alternative to A Levels, Applied Generals or an Apprenticeship.

The typical student will likely have:

- a clear idea as to the industry sector they wish to pursue as a career
- an idea of the type of job role they'd like to explore as a career
- taken an active choice not to pursue an Apprenticeship (either due to lack of availability or the wish to remain in full time education).

This Technical Qualification aligns to the Software Development Technician Level 3 Apprenticeship and therefore supports progression to entry-level job opportunities in software design and development. Job roles could include:

- Software Development Technician
- Junior Developer
- Junior Web Developer
- Junior Application Developer
- Junior Mobile App Developer
- Junior Games Developer
- Junior Software Developer
- Junior Application Support Analyst
- Junior Programmer
- Assistant Programmer
- Automated Test Developer.

The jobs available to the students will be based on their individual abilities in the digital sector and will be supported by their achievement of this qualification.

Alternatively, students could progress sideways to the Level 3 Software Technician Apprenticeship to develop and gain certification of their occupational competence, or they could progress to higher level Apprenticeships such as the Level 4 Software Developer, depending on their skills or experience.

Where students may not have access to an Apprenticeship or would prefer a more academic route, they could progress to relevant Higher National Certificate (HNC) or Higher National Diploma (HND) programmes or digital degree programmes such as Computer Games Programming BCs, Software Engineering BSc, Virtual Reality Design BA, Computing BSc, Digital Media Design and Development BSc or Computer Science BSc.

Students should always check the entry requirements for each degree programme with the relevant higher education provider.

Prior learning requirements

There are no formal prior learning requirements for the *T Level Technical Qualification in Digital Production, Design and Development*.

However, as a Provider, it is your responsibility to ensure the students you recruit have a reasonable expectation of success on the programme. Formal entry requirements are not set by Pearson, but we expect students to have qualifications at or equivalent to Level 2.

Students are most likely to succeed if they have:

- five GCSEs/international GCSEs at grade 4 or above, including English, Maths and Science, and/or
- Vocational Tech Award qualification(s) at Level 2 at Pass and above in a relevant subject, e.g. BTEC Tech Award in Digital Information Technology.

Students may demonstrate the ability to succeed in various ways. For example, they may have relevant work experience or specific aptitude shown through diagnostic tests or non-educational experience.

Transferring between T Levels

We expect some students to switch between T Levels. During Year 1, Providers should consider the degree of overlap between the two T Levels and the remaining time pre-assessment, to determine if transfers should be permitted. For funding purposes, it is important that students have made a decision about their T Level and occupational Specialism by the end of their first year. T Level Core assessments will vary in terms of content coverage, duration, and method, and therefore attainment from one T Level cannot count towards another.

What does the qualification cover?

The Technical Qualification content has been designed from the Outline Content created by the Institute for Apprenticeships and Technical Education and the Digital T Level panel.

We have used the Outline Content to create the Technical Qualification specification and assessment, which has been validated by our own panel of digital employers and education providers to ensure it is appropriate for the progression routes identified in the above section.

Students will learn about the following topics:

- problem solving
- programming
- emerging issues and impact of digital
- legislation and regulatory requirements
- business context
- data
- digital environments
- security.

3. General Competency Frameworks for T Levels

The General Competency Framework for T Levels articulates English, mathematical and digital competencies that students are required to develop over the course of the qualification. The tables below list the competencies from the framework that are relevant to the *T Level Technical Qualification in Digital Production, Design and Development*.

Competencies that can be developed in relation to a specification element of content are referenced in the column next to this content element. These competencies should be delivered through the content of this qualification and tutors should seek opportunities to allow students to develop the relevant skills to enable them to reach threshold competence in the specialism.

The English, Maths and Digital competencies are embedded in both the Core Component and the Occupational Specialist Components of the *T Level Technical Qualification in Digital Production, Design and Development*. This is so that students are able to demonstrate their knowledge and understanding of these skills over the course of the qualification.

General English competencies

E1	Convey technical information to different audiences
E2	Present information and ideas
E3	Create texts for different purposes and audiences
E4	Summarise information/ideas
E5	Synthesise information
E6	Take part in/leading discussions

General maths competencies

M1	Measure with precision
M2	Estimate, calculate and spot errors
M3	Work with proportion
M4	Use rules and formulae
M5	Process data
M6	Understand data and risk
M7	Interpret and represent with mathematical diagrams
M8	Communicate using mathematics
M9	Cost a project
M10	Optimise work processes

General digital competencies

Students should be supported to develop the digital knowledge and skills needed in order to:

D1	Use digital technology and media effectively
D2	Design, create and edit documents and digital media
D3	Communicate and collaborate
D4	Process and analyse numerical data
D5	Be safe and responsible online
D6	Code and program

4. Core Component

Content summary

The core content covers the knowledge, understanding and application of contexts, concepts, theories and principles relating to the following areas:

1. Problem solving

2. Introduction to programming

3. Emerging issues and impact of digital

4. Legislation and regulatory requirements

5. Business context

6. Data

7. Digital environments

8. Security

Paper 1: Digital Analysis, Legislation and Emerging Issues

Content area 1: Problem solving

Students must be able to apply problem-solving skills to analyse problems and to identify solutions that can be developed into computer programs. Students will be expected to solve realistic problems that may form a complete solution or a sub-part of a larger program. Students will be expected to use the flowchart symbols and pseudocode listed in Appendix 1.

What students need to learn		
1.1 Computational thinking		
1.1.1	Be able to use top-down, bottom-up and modularisation approaches to solve problems.	**E1** **M7**
1.1.2	Be able to decompose problems by: • identifying and describing the main features of a problem or process • breaking a problem down into smaller, more manageable parts.	**E1 E2** **M7**
1.1.3	Be able to use pattern recognition to: • identify and describe trends and similarities within and between problems and processes • identify and describe common features between a given problem and existing solutions • make predictions and assumptions based on identified patterns.	**M2 M4 M8** **D4**
1.1.4	Be able to use abstraction to: • identify information that is needed to solve an identified problem • filter out unnecessary details at different stages of a problem • create a layer of abstraction appropriate to the stage in the problem-solving process, including: • what inputs are needed • what the expected outputs are • things that will vary • things that will remain constant • key actions the program must perform • repeated processes the program will perform.	**E1 E5** **M10**

What students need to learn		
1.2 Algorithms		
1.2.1	Understand what algorithms are and how they are expressed (flowcharts, written descriptions, pseudocode, program code).	
1.2.2	Be able to express an algorithm using flowcharts and pseudocode, and understand how to use these when planning a digital solution.	**M7 M10 D3 D6**
1.2.3	Be able to write algorithms that make use of programming constructs (sequence, selection, iteration).	**M7 M10 D3 D6**
1.2.4	Understand the purpose of a given algorithm (flowcharts, written descriptions, pseudocode, program code) and how the algorithm works.	**M4 M7**
1.2.5	Be able to determine the correct output of an algorithm.	**M2 M4 M5 D4**
1.2.6	Be able to identify and correct errors in an algorithm (flowcharts, written descriptions, pseudocode, program code).	**M2 M4 M5 M7 D4**

Content area 2: Introduction to programming

Students should be able to apply an understanding of computer programming to solve problems. Students should be able to design, read, write and debug program code. Students will be expected to solve realistic problems that may form a complete solution or a sub-part of a larger program.

When designing a program, students will be expected to use the flowchart symbols listed in Appendix 1. Students will be expected to write, interpret and debug code in the programming language Python 3.

Students will be expected to create functions and procedures to structure and carry out programming requirements.

Students will be expected to use code development tools, including Integrated Development Environments (IDE).

When writing, interpreting and debugging code, students will be expected to understand and use the libraries, functions and methods listed in Appendix 1.

What students need to learn		
2.1 Program data		
2.1.1	Understand the use of, and need for, data types: • string • character • integer • real/float • Boolean.	**M5** **D6**
2.1.2	Be able to declare and use constants and variables that use appropriate data types.	**M4 M5** **D4 D6**
2.1.3	Understand the use of, and need for, data structures: • list • array • dictionary.	**M4 M5 M6** **D4 D6**

2.1.4	Understand how to manage variables within a program, including:
	• use of local and global variables
	• when local and global variables should be used, and why
	• variable naming conventions:
	  ○ meaningful names
	  ○ case (e.g. camelCase, UPPER CASE)
	  ○ underscores (e.g. address_line_1).
		M4 M5 M6
		D4 D6

What students need to learn		
2.2 Operators		
2.2.1	Understand the purpose of, and how to use, mathematical operators in program code and algorithms (add, subtract, divide, multiply, integer division, modulus).	**M3 M4**
2.2.2	Understand the purpose of, and how to use, relational operators (==, <, >, <>, <=, >=).	**M3 M4**
2.2.3	Understand the purpose of, and how to use, Boolean operators (NOT, AND, OR).	**M4** **D6**
2.3 File handling		
2.3.1	Understand how to use text files for input and output of data.	**M5** **D1 D4 D6**
2.4 Program structure		
2.4.1	Understand how sequence, selection (branching) and iteration are used within programs and algorithms.	
2.4.2	Be able to write, interpret and debug code that makes use of sequence: • Determine the most efficient and logical order for actions within a process. • Understand the correct order of operations in calculations and processes, to ensure outputs are accurate and errors are avoided.	**M2** **D6**
2.4.3	Be able to write, interpret and debug code that makes use of selection (branching): • IF, THEN, ELSE, ELSEIF (ELIF) • CASE.	**M2 M4 M10** **D6**
2.4.4	Be able to write, interpret and debug code that makes use of iteration: • Understand how 'For' loops are used to iterate code a set number of times. • Understand how 'While' loops are used to iterate code while a set criterion is met. • Understand how loops are used to iterate code until a set criterion is met.	**M4 M10** **D6**
2.4.5	Be able to declare and call functions and procedures.	**M4 M10** **D6**
2.4.6	Understand how standard searching and sorting algorithms work, and the benefits and drawbacks of each: • linear and binary search • bubble sort, insertion sort, merge sort.	**E1** **M4 M5** **D4 D6**

What students need to learn		
2.5 Built-in functions		
2.5.1	Understand the benefits and drawbacks of using pre-written code.	**E4 D6**
2.5.2	Be able to select and justify the use of pre-written code provided by the Python programming language (e.g. built-in functions, standard libraries).	**E4 E5 D6**
2.5.3	Be able to write code that makes use of user-written and pre-written code (e.g. built-in functions, standard libraries).	**E4 E5 D6**
2.6 Validation and error handling		
2.6.1	Understand the need for different types of input validation and be able to write, interpret and debug code that makes use of these validation techniques: • presence check • length check • type check • format check • range check • check digit.	**M2 M4 M5 M6 D6**
2.6.2	Understand the need to develop reliable and robust code.	**M6 D6**
2.7 Maintainable code		
2.7.1	Understand the accepted style conventions (such as Python's PEP 8) and how these are implemented to create readable and maintainable code.	**M4 D3 D6**
2.8 Testing		
2.8.1	Understand the fundamental importance of testing for all components: • software • hardware • data • interfaces • resulting service (final product).	**D1 D6**

What students need to learn		
2.8 Testing *continued*		
2.8.2	Understand the use of testing and quality assurance methodologies to seek out problems and issues: • concept testing • unit testing • boundary testing • integration testing • performance testing • system testing • acceptance and usability testing • regression testing • load/stress testing.	**E5** **M6** **D1 D6**
2.8.3	Understand how automated and functional testing tools can be applied to test digital systems and code.	**E5** **D1 D6**
2.8.4	Understand how to apply root cause analysis to solve problems: • what it is (the five whys) • when to use it • how to use it • what next.	**E5** **D3**
2.8.5	Understand how to construct an effective test plan, including: • identifying tests to be carried out • describing the purpose of the identified test • identifying test data to be used (valid, valid extreme, invalid, invalid extreme, erroneous) • describing the expected results.	**E1 E5** **M2** **M10** **D2 D6**

Content area 3: Emerging issues and impact of digital

Students should be able to apply an understanding of ethical and moral issues in the digital sector in a range of business contexts. They should explore how developments in technology impact on organisations, individuals and society as a whole.

Students should be aware of the ever-developing nature of digital technologies, and keep up to date with knowledge of important and innovative developments in the sector.

What students need to learn		
3.1 Moral and ethical issues		
3.1.1	Understand the ethical and moral issues that an increasing reliance on technology raises, and how organisations and individuals can respond to these challenges: • acceptable use • autonomous operation • changes in societal norms and the behaviour of individuals • changes in the culture within an organisation • environmental issues • globalisation • inclusion and diversity • monitoring of employees • open source and Creative Commons • the collection and use of data • unequal access to technology and/or digital services.	**E2 E4 E5** **D5**
3.1.2	Understand how organisations and individuals respond to ethical and moral issues when designing and developing digital systems, including: • use of guidelines from professional organisations • strategic planning and decisions • the content of internal policy documents • company culture and how this is established, communicated and sustained • whistleblowing.	**E2 E4 E5** **D5**
3.1.3	Understand how individuals use a range of observational techniques to inform situational awareness: • observing normal behaviour • awareness of co-workers • recognising changing or abnormal behaviour.	**E2 E4 E5** **D5**
3.2 Emerging trends and technologies		
3.2.1	Understand how developments in digital technologies impact on organisations, individuals and society, including: • Internet of Things (IoT) • Artificial Intelligence (AI), machine learning and deep learning • Augmented Reality (AR) and Virtual Reality (VR).	**E2 E4 E5** **D1**

Content area 4: Legislation and regulatory requirements

Students should be able to apply an understanding of legal issues in the digital sector in a range of business contexts. Students should explore how compliance with legislation impacts on the way in which organisations and their stakeholders use and interact with digital technologies.

Students should be aware of the ever developing nature of digital technologies and keep up to date with changes in legislation in response to technological developments.

What students need to learn		
4.1 Legislation		
Understand the role of current legislation and its impact on the design, development and use of digital in relation to:		
4.1.1	Health and safety when working with computers: • display screen regulations • general working environment • possible risks and prevention.	**E1 E2 E4 E5 D1**
4.1.2	Data Protection Act: • the principles of the act • General Data Protection Regulations (GDPR) • marketing consent • the rights of the data subject • enforcement.	**E1 E2 E4 E5 M6 D1 D5**
4.1.3	Computer Misuse Act: • the principles of the act • consequences (company and employee) • employee awareness.	**E1, E2, E4, E5 D1, D5**
4.1.4	Equality Act: • types of discrimination (protected characteristics) • where individuals are protected • when to take action against discrimination. • how individuals can be discriminated against (direct, indirect, harassment and victimisation).	**E1 E2 E4 E5 D1 D5**
4.1.5	Intellectual Property Act: • unregistered designs • registered designs • patents.	**E1 E2 E4 E5 D1 D5**

What students need to learn		
4.1 Legislation *continued*		
4.1.6	Understand the use of digital technologies for monitoring the workplace: • monitoring electronic communications • use of secret monitoring • employers' monitoring policies • monitoring systems.	**E1 E2 E4 E5 D1 D5**
4.1.7	Understand the role of legislation relating to international law and its importance when designing, developing and using digital systems.	**E1 E2 E4 E5 D1 D5**
4.2 Guidelines and codes of conduct		
4.2.1	Understand the purpose and role of codes of conduct produced by professional bodies for the use of digital: • British Computer Society (BCS) Code of Conduct • The Institution of Analysts and Programmers Code of Conduct.	**E1 E2 E4 E5 D1 D5**
4.2.2	Understand the guidelines provided in professional codes of practice in terms of: • professional responsibilities (quality of work, meeting deadlines, communication, confidentiality, trust) • contribution to society • safety • security and privacy • innovation.	**E5 D5**
4.2.3	Understand the impact that implementing guidelines from professional codes of practice has on organisations and their stakeholders.	**E4 E5 D5**
4.2.4	Understand how guidelines and agreed standards ensure the accessibility and quality of IT systems, including: • ISO (international Standards Organisation) standards • Web Content Accessibility Guidelines (WCAG) 1.0 and 2.0 • World Wide Web Consortium (W3C®) • Internet Engineering Task Force (IETF).	**E4 E5 D1 D5**
4.2.5	Understand the role and implications of acceptable use policies within an organisation.	**E4 E5 D5**

Paper 2: The Business Environment

Content area 5: Business context

Students must apply an understanding of the business environment including the importance of serving customer and end user, business needs, stakeholders such as customers, competitors, suppliers and government and the social, political, legal and technological factors drive the need for and use of digital skills technologies.

What students need to learn		
5.1 The business environment		
5.1.1	Understand the purpose of different types of organisations in a range of sectors: • to provide a service • to provide a product.	**E2 E4**
5.1.2	Understand the key areas of organisations and how IT is used to support them: • Human Resources • Research, Design and Development • Logistics • Marketing • Finance • Management.	**E2 E4 E5**

5.1.3	Understand how digital supports the business needs of organisations.
	• the use of digital to enable automated stock/inventory control:
	  ○ how software is used
	  ○ how hardware is used
	  ○ the processes carried out
	  ○ how different parts of the system communicate with each other.
	• the use of traditional and cloud-based technologies and services to communicate and collaborate with internal and external stakeholders and facilitate collaboration.
		E4 E5
		D1 D3

5.1.4	Understand the factors that determine the feasibility of a digital project: • benefits and drawbacks • risks, constraints and dependencies.	**E5** **M2** **M8** **M9** **D4**

What students need to learn		
5.1 The business environment *continued*		
5.1.5	Understand how digital is used to meet user needs and ensure quality of product/service: • appropriate and effective functionality • reduction of pain points • accessibility considerations • compatibility • availability • good user experience • cultural awareness and diversity.	**E5** **D1**
5.1.6	Understand how the characteristics of end users affect the use and characteristics of digital technologies to access a service or product: • age • skills • education level • internal/external audience • level of technical knowledge • additional needs (e.g. users with sight or hearing loss).	**E5** **D1 D2**
5.2 Digital value to business		
5.2.1	Understand the importance of digital within organisations, and the ways in which digital is used to add value to a company: • engagement of customers, users and other stakeholders • provision of products and services to customers • measurable value (reducing overheads, improving efficiency, facilitating growth, recruiting talent) • supporting processes and business models (product design, manufacturing control, data modelling, local and remote working) • context and market environment (stakeholders, user profiling, personalised/appropriate content, data).	**E5** **D1**

What students need to learn		
5.3 Technical change management		

5.3.1	Understand the factors that trigger change in organisations:
	• Planned for factors
	  ○ adding additional features and/or services
	  ○ diversification
	  ○ scaling
	  ○ rebranding
	  ○ adoption of new technologies
	  ○ changes in legislation
	  ○ response to competition.
	• unforeseen or previously unpreventable factors
	  ○ crisis (natural disasters, terrorism, cyber attacks)
	  ○ zero-day vulnerabilities
	  ○ data corruption
	  ○ system failures.
		E2 E4 E5

5.3.2	Understand the technical change management process, including: • identifying the changes to be made • identifying and communicating potential risks and desired impact(s) to stakeholders • configuration of the new system or process • method of implementing change (parallel, phased, direct, pilot) • documenting the change process • importance of rollback planning • importance of ensuring reproducibility of performance and outcome • traceability of requirements throughout the development lifecycle.	**E1 E5 M6 M9 D3**
5.3.3	Understand how organisations respond to, prepare for, manage and reinforce change (relevant to digital) in a range of contexts: • economic, banking and financial • environmental • legal • people • political • regulatory • social • technological.	**E1 E5 M6 M9 D3**

What students need to learn		
5.3 Technical change management *continued*		
5.3.4	Understand the benefits and drawbacks of technical change in organisations in relation to: • productivity • communication • security • replacing existing products • updating or changing processes • support for stakeholders • costs • stakeholder experience • company reputation.	**E1 E4** **M2 M6** **M9**
5.4 Risks in a business context		
5.4.1	Understand the potential risks to organisations of use of digital systems and technologies: • security breaches • privacy breaches • regulatory and legal non-compliance • system failure • audience exclusion • emerging rival services (mobile devices, digital download and cloud services) • rapid changes in technology and trends (education and transport sectors).	**E1 E5** **M6** **D5**
5.4.2	Understand the potential impact of identified risks on the organisation and its stakeholders.	**E1 E5** **M6** **D5**

Content area 6: Data

Students must apply an understanding of the use of data by organisations to support business needs. They should explore the benefits and challenges that digital technologies present in terms of the creation and use of data.

What students need to learn		
6.1 Data and information in organisations		
6.1.1	Understand the differences and links between data, information and knowledge.	**M6** **D4**
6.1.2	Understand why organisations need data and information and how they are used: • analysing market trends to identify patterns which inform decisions • system performance analysis • user monitoring • targeted marketing • informed decision making (strategic, tactical and operational) • threat/opportunity assessment (break-even, predictive models, cost analysis, market trends).	**E5** **M6** **D4**
6.1.3	Understand how data is generated: • human generated • artificial intelligence/machine learning • sensors • Internet of Things (IoT) • transactional data.	**E5** **M6** **D4**
6.2 Data formats		
6.2.1	Understand the forms that data can take and the implications this has on use and analysis: • data types (date, integer, real, character, string, Boolean) • common forms of data format (JSON, fixed-width text file, CSV, ASCII, XML).	**E5** **M5 M6** **D4 D6**
6.2.2	Understand the difference between file-based and directory-based structures, and how they are used in data analysis.	**E5** **M5 M6** **D4 D6**

What students need to learn		
6.3 Data systems		
6.3.1	Understand the features and functions of data systems and their importance to organisations: • data wrangling (structure, clean, enrich, validate, output) • core functions (input, search, save, integrate, organise (index), output, feedback loop) • data entry and maintenance (online data entry, risk of data entry errors, time to create the entry screen and enter data) • visualisation (graphs/charts, data tables, reports, infographics).	**E5 M5 M6 D4 D6**
6.3.2	Understand the purpose of business information tools and their use in business (e.g. business intelligence software, financial planning and analysis, Customer Relationship Management (CRM)).	**E5 M5 M6 D4 D6**
6.3.3	Understand the features of different data models and how organisations use them to organise data: • conceptual data model • logical data model • physical data model • hierarchical database model • relational model.	**E5 M5 M6 D4 D6**
6.4 Data management		
6.4.1	Understand factors that determine how data is gathered, entered and maintained: • the 'Six Vs' (volume, variety, variability, velocity, veracity, value) • data assurance/quality (validation, verification, reliability, redundancy) • types of data • research population • qualitative data and quantitative data • legislation and regulatory compliance • ethics • organisational factors (time, skills, cost).	**E5 M5 M6 D4 D6**
6.4.2	Understand the purpose of data analysis tools and their use in business (data warehousing, data lakes, data mining, reporting).	**E5 M5 M6 D4 D6**
6.4.3	Understand the role of metadata classification in defining the meanings of data.	**E5 M5 M6 D4**

What students need to learn		
6.4 Data management *continued*		
6.4.4	Understand the use of data/access entitlements/permissions management, and its impact on organisations and stakeholders: • authorisation • privileges • access rights • rules.	**E5** **M5 M6** **D4 D6**
6.4.5	Understand how data can be accessed and managed across different platforms: • the role and use of Application Programming Interfaces (APIs) in managing, accessing and using data.	**E5** **M5 M6** **D4 D6**
6.4.6	Understand the concepts of data at rest, data in use and data in motion, and when each is used.	**E5** **M5 M6** **D4**

Content area 7: Digital environments

Students should be able to apply an understanding of the different platforms of delivery that enable access to digital tools and services. They should explore how different digital environments meet the needs of organisations and their stakeholders. They must apply an understanding of digital environments in a range of business contexts.

What students need to learn		
7.1 Physical environments		
7.1.1	Understand the features and characteristics of different types of physical computer system: • personal computers • mobile devices • servers • smart/internet-enabled devices.	**E1 E2 E4 E5** **D1**
7.1.2	Understand the features and characteristics of hardware and peripherals used in physical computer systems: • input devices • output devices • processors • memory • secondary storage devices (internal and external) • motherboard/mainboard • cooling • sensors.	**E1 E2 E4 E5** **D1**

7.1.3	Understand the purpose and functions of software used in computer systems:
	• operating systems
	  ○ batch operating system
	  ○ multitasking/time-sharing operating system
	  ○ real-time operating system
	  ○ network operating system
	  ○ mobile operating system
	• utility software
	• application software
	• code development tools (IDEs, debuggers).
		E1 E2 E4 E5
		M6
		D1

7.1.4	Understand the benefits and drawbacks of software, hardware and peripherals in different contexts.	**E1 E2 E4 E5** **M6** **D1**

What students need to learn		
7.1 Physical environments *continued*		

7.1.5	Understand how physical data storage and recovery systems work, their features, benefits and drawbacks:
	• redundant array of independent disks (RAID)
	  ○ RAID 1
	  ○ RAID 5
	  ○ RAID 10
	• network attached storage (NAS)
	• storage area network (SAN).	

7.2 Networks		
7.2.1	Understand the benefits and drawbacks of connecting devices to form networks.	**E1 E2 E4 E5 M6 D1**
7.2.2	Understand the features, characteristics, benefits and drawbacks of wireless connection methods.	**E1 E2 E4 E5 M6 D1**
7.2.3	Understand the features, characteristics, benefits and drawbacks of wired connection methods.	**E1 E2 E4 E5 M6 D1**
7.2.4	Understand different types of network: • LAN • WAN • PAN.	**E1 E2 E4 E5 M6 D1**
7.2.5	Understand the concepts of bandwidth and latency, and their effect on the performance of networks and connected systems.	**E1 E2 E4 E5 D1**
7.2.6	Understand the concept of different network models: • client-server • thin client • peer-to-peer.	**E1 E2 E4 E5 M6 D1**
7.2.7	Understand the characteristics of network topologies: • logical vs physical • star • mesh • tree • VLAN.	**E1 E2 E4 E5 D1**

What students need to learn		
7.2 Networks *continued*		
7.2.8	Understand the role and characteristics of common components of a network: • server • internet connection/internet backbone • router • network switch • client.	**E1 E2 E4 E5 D1**
7.2.9	Understand the seven-layer OSI model to describe how applications communicate over a network, including the function and related protocols of each layer: • application layer • presentation layer • session layer • transport layer • network layer • data link layer • physical layer.	**E1 E2 E4 E5 M4 M6**
7.2.10	Understand the four-layer TCP/IP model to describe how applications communicate over a network, including the function and related protocols of each layer: • application layer • transport layer • internet layer • network access layer.	**E1 E2 E4 E5 M4 M6**
7.2.11	Understand the role of data packets in transmitting over a network, including: • contents and structure of a data packet • role of the components of a data packet • packet switching error handling – Cyclic Redundancy Check (CRC).	**E1 E2 E4 E5 M4 M6 M10**
7.2.12	Understand the role of common network protocols.	**E1 E2 E4 E5 M4 M6**
7.2.13	Understand how physical, virtual and cloud environments, along with networks, are used in combination, including the Internet of Things (IoT), to solve problems and meet the needs of organisations and their stakeholders.	**E1 E2 E4 E5 M10 D1 D3**

7.3 Virtual environments		
7.3.1	Understand the key features of virtual environments: • increased security • managed execution • sharing • aggregation • emulation • isolation • portability.	**E1 E2 E4 E5 M10 D1 D3**
7.3.2	Understand the benefits and drawbacks of the use of virtual environments for organisations in a range of contexts.	**E1 E2 E4 E5 M10 D1 D3**
7.4 Cloud environments		
Understand the ways in which organisations use cloud environments to provide access to digital tools, services, storage and systems.		
7.4.1	Understand the concepts of cloud computing deployment in terms of: • applications • data • runtime • middleware • operating system • virtualisation • servers • storage • networking.	**E2 E4 E5 M6 D1 D3 D4**

7.4.2	Understand common cloud delivery models and the way in which responsibility and ownership of resources are distributed between the subscriber and service provider:
	• IaaS (Infrastructure as a Service)
	  ○ subscriber (applications, data, runtime, middleware, operating system)
	  ○ service provider (virtualisation, servers, storage, networking)
	• PaaS (Platform as a Service)
	  ○ subscriber (applications, data)
	  ○ service provider (runtime, middleware, operating system, virtualisation, servers, storage, networking)
	• SaaS (Software as a Service)
	  ○ subscriber (user only)
	  ○ service provider (applications, data, runtime, middleware, operating system, virtualisation, servers, storage, networking).
		E2 E4 E5 M6 D1 D3 D4

What students need to learn		
7.4 Cloud environments *continued*		
7.4.3	Understand the concept of DaaS (Data as a Service) in terms of: • data science platforms • dashboards • business information tools • data lakes • databases • file systems.	**E2 E4 E5 M6 D1 D3 D4**
7.4.4	Understand how DaaS (Data as a Service) is used by organisations, and the benefits and drawbacks it provides for organisations and their stakeholders.	**E2 E4 E5 M6 D1 D3 D4**
7.4.5	Understand the concept of cloud sourcing and cloud portability, and the implications for service providers and organisations (subscribers).	**E2 E4 E5 M6 D1 D3 D4**
7.5 Resilience of environment		
7.5.1	Understand the need to ensure digital environments are resilient, and the impact on organisations and stakeholders if this is not achieved.	**E2 E4 E5 M6 D1 D4**
7.5.2	Understand methods used to improve the resilience of digital environments: • data and system redundancy • back-up systems • hot, cold and warm sites • data back-up and recovery procedures • device hardening	**E2 E4 E5 M6 D1 D4 D6**
7.5.3	Understand the benefits and drawbacks of methods used to improve the resilience of digital environments.	**E2 E4 E5 M6 D1 D4 D6**

Content area 8: Security

Students should be able to apply an understanding of the potential risks posed by the use of digital to an organisation and its stakeholders. Students should explore established and emerging risks, and understand ways in which risks can be mitigated. They should be able to demonstrate an understanding of risks and mitigation measures in a range of business contexts.

What students need to learn		
8.1 Security risks		
8.1.1	Understand the importance of maintaining privacy and confidentiality of an organisation's information, as well as that of stakeholders, including: • information about salaries • employee perks • client lists • trade secrets • sales numbers • customer information • news about pending restructuring.	**E2 E4 E5** **M6** **D1 D5**
8.1.2	Understand the potential impact on an organisation of failing to maintain privacy and confidentiality.	**E2 E4 E5** **M6** **D1 D5**
8.1.3	Understand potential technical threats and vulnerabilities to systems, data and information, including: • botnets • distributed denial-of-service (DDoS) • hacking • malware (including ransomware) • social engineering (pharming, phishing) • insecure Application Programming Interfaces (APIs) • use of ad hoc or open networks • eavesdropping/man-in-the-middle attacks.	**E2 E4 E5** **M6** **D1 D5**
8.1.4	Understand potential physical vulnerabilities to systems, data and information, including: • location of system or asset • circumstances of use • characteristics of users/community • system or asset layout • system or asset design/robustness.	**E2 E4 E5** **M6** **D1 D5**

What students need to learn		
8.1 Security risks *continued*		
8.1.5	Understand potential **human** threats and vulnerabilities to systems, data and information, including: • human error • malicious employees • disguised criminals • targeted attack.	**E2 E4 E5** **M6** **D1 D5**
8.2 Threat mitigation		
8.2.1	Understand the concept of the CIA (confidentiality, integrity, availability) and how it can be applied to define security aims.	**E2 E4 E5** **M6** **D1 D5**
8.2.2	Understand the interrelationship between security, identity, confidentiality, integrity, availability, threat, vulnerability and risk management within a business context.	**E2 E4 E5** **M6** **D1 D5**

8.2.3	Understand processes and procedures to mitigate threats and ensure security, including:
	• air gapping
	• anti-virus and anti-malware programs
	• certification of APIs
	• configuration and management of software-based access control
	• device hardening
	• encryption (hashing, asymmetric, symmetric)
	• user access restrictions
	  ○ usernames, passwords and passphrases
	  ○ data access levels/permissions
	  ○ physical access control/restrictions
	• multi-factor authentication (possession-based, biometric, knowledge, location-based)
	• firewalls
	• password managers
	• policy, policy enforcement and training
	• SYN cookies
	• use of Virtual Private Networks (VPNs)
	• security testing (penetration testing, white/grey hat hackers).
		E2 E4 E5
		M6
		D1 D5

Core project

Task 1

Planning a project		
Project planning tools	Be able to use project planning tools to apply understanding of project planning in response to a scenario.	
Gantt chart	a) Assess the strengths and skills of people and assign appropriate tasks to them. b) Make scheduling decisions in response to a defined deadline. c) Prioritise activities or tasks based on analysis of requirements. d) Demonstrate how to correctly and appropriately assign resources to project tasks. e) Produce a Gantt chart to show project tasks and organise them efficiently, using an appropriate Software Development Lifecycle model.	**E5** **M1** **D1**
Resource and cost plan	a) Identify and calculate costs of a project, including: • materials • physical resources • personnel. b) Select and allocate resources to the resource list, and correctly attribute costs to provide an accurate estimate of the total project cost.	**M8**
Rationale	a) Consider the factors that are most relevant when planning projects. b) Justify project planning decisions made, with consideration given to: • cost, risk and benefits to identified stakeholders • order and timing of tasks • selection and allocation of resources, including personnel and physical resources c) dependencies and prerequisites.	**E5** **M9**

Task 2

Identifying and fixing defects in an existing code		
Use of testing to identify defects	a) Assess the given code against requirements. b) Carry out testing to identify issues in given code. c) Perform any remedial actions required, justifying any decision made when fixing the defect.	
Documenting the testing process	Provide annotated evidence of testing, including: • identifying tests to be carried out • describing the purpose of the identified test • identifying test data to be used (valid, valid extreme, invalid, invalid extreme, erroneous) • describing the expected results • describing the actual results of the tests performed • comparing the actual results of testing with the expected results • describing any further actions that are required • refining the system as required.	**E1 E2** **M10** **D3**
The solution	a) Correct errors in code add and/or remove code to ensure that the given code is functional and meets the given requirements. b) Follow appropriate programming conventions when fixing code to ensure that it makes use of precise logic and programming structures throughout, so that the program produces consistently correct outcomes.	**M4** **M5** **M7**

Task 3

Designing a solution		
Decomposition of the problem	a) Break down the problem into smaller parts suitable for computational solutions, justifying any decisions made. Make effective use of detailed abstraction and refinement. b) Use elements of reusable components. c) Use good decomposition to show all the necessary subsystems that make up the main solution. d) Visualise the decomposition. e) Use appropriate tools for communicating algorithms (e.g. flowcharts, pseudocode, data flow diagrams).	**D4**
Application of logical thinking	a) Describe the parts of the solution using algorithms, justifying how these algorithms form a complete solution to the problem. b) Clearly define the steps. c) Uniquely define each step. They should depend on the input and the result of the preceding steps. d) Ensure the algorithm makes use of key constructs (e.g. sequence, selection and iteration).	**M2** **D2**
Use of conventions	a) Demonstrate correct use of structure and convention for the chosen method of communication (e.g. flowcharts, pseudocode, data flow diagrams), such as correct use of symbols for flowcharts and key words used in pseudocode. b) Select and make consistent use of appropriate naming conventions throughout.	**E3**
Communication of the design	a) Ensure design documents are of sufficient detail to: • effectively communicate the intended solution • allow the client to make informed decisions • allow a third party to use design documents to create the proposed solution. b) Communicate intended solution effectively and clearly, with use of: • appropriate combination of written and diagrammatical presentation • appropriate use of technical vocabulary • consideration of audience • explanations of structures and process in the design.	**E4** **D3**

Task 4a

Developing a solution		
The solution	Apply an undertaking of programming to develop a solution that meets the requirements of a brief, including: • refining the system as required • demonstrating an appropriate level of technical skill, and understanding of programming techniques and problem solving • use of pre-written and user-written modules with appropriate interface • discussion of any issues that may have come about from the testing.	
Code organisation	Ensure code produced for the solution is appropriate to meet the demands of the brief, including: • trying to avoid multiple pages of nested if-clauses and for-loops with a lot of copy-pasted procedural code • clear meaningful indentation • pieces of logic, classes or objects, with proper structure • comments whenever possible to help explain the logic • good use of local variables and minimal use of global variables • use of constants • well-designed interface • consistent style throughout • defensive programming • good exception handling.	**M10**
User experience	Meet user needs, with consideration of: • consistency of the product • accessibility to all types of users • user interface design (visual design in line with brand goals) • meaningful feedback, changes made and problems overcome.	**E6**
Responsive-ness to the brief	a) Formal documentation of testing is not required in this activity. b) Testing by the student should still be carried out but its use should be implicit through: • a working product • a product that meets the requirements detailed in the given task brief.	**E6**

Task 4b

Reflective evaluation		
Programming outcomes	a) Be able to apply reflection and evaluation techniques. b) Provide evidence that the product meets brief requirements: • include measures against success criteria • provide evidence that the product meets user needs from testing • discuss how it could be improved if the problem was revisited and given detailed consideration.	
Comparison to designs	• Provide a well-developed line of reasoning which is clear and logically structured. The information presented is relevant and substantiated. • Discuss the maintainability of the solution. • Discuss potential further developments of the solution. • Discuss any changes and why they are different from the original designs.	**E3**

Resources for the delivery of the Core Component content

The following resources will be required for the delivery of the Core Component for this technical qualification:

- Python 3 and the appropriate IDE
- Python Libraries, in addition to standard libraries or other libraries that provide the same/comparable functionality/capabilities:
  - pandas
  - Tkinter
  - wxPython
  - NumPy
  - TensorFlow
  - Matplotlib.

Scheme of Assessment – Core Component

There are three assessments in the Core Component of the *T Level Technical Qualification in Digital Production, Design and Development*:

1. Examination Paper 1: Digital Analysis, Legislation and Emerging Issues
2. Examination Paper 2: The Business Environment
3. Employer Set Project.

Core examinations

Paper 1: Digital Analysis, Legislation and Emerging Issues
Written examination: 2 hours 30 minutes **33.33% of the core assessments** **100 marks**
Content overview **1.** Problem solving **2.** Introduction to programming **3.** Emerging issues and impact of digital **4.** Legislation and regulatory requirements
Assessment overview • An externally-assessed written examination comprising two sections. Students answer all questions in Section A and Section B. • The examination will include short, medium and extended open-response questions, as well as labelling questions. • The examination will be set and marked by Pearson.

Paper 2: The Business Environment
Written examination: 2 hours 30 minutes **33.33% of the core assessments** **100 marks**
Content overview **5.** Business context **6**. Data **7.** Digital environments **8.** Security
Assessment overview • An externally-assessed written examination comprising two sections. Students answer all questions in Section A and Section B. • The examination will include short, medium and extended open-response questions, as well as labelling questions. • The examination will be set and marked by Pearson.

Both examinations will follow the same paper structure but they will assess different core content, and will be available paper-based. There are two sections in each paper:

- Section A is weighted 40%.
- Section B is weighted 60%.

Core examination Assessment Objectives

Assessment Objective			Range for Papers 1 and 2	Paper 1 proportion	Paper 2 proportion
AO1	**1a (i)**	**Knowledge** (isolated knowledge)	0%–3%	6%	6%
	1a (ii)	**Knowledge** (embedded knowledge)	3%–6%		
	1b	**Understanding**	N/A	25%	28%
AO2		**Application**	N/A	48%	45%
AO3		**Analyse and Evaluate**	N/A	21%	21%

Employer Set Project

Core project – Employer Set Project
Externally assessed project: 14.5 hours **33.33% of the core assessments** **100 marks**
Content overview When responding to the core project, students will need to draw on knowledge and understanding form across the core content in a synoptic manner, in order to effectively respond to a brief within a vocational context.
Assessment overview There are five parts to the assessment: Task 1: Planning a project Task 2: Identifying and fixing defects in existing code Task 3: Designing a solution Task 4a: Developing the solution Task 4b: Reflective evaluation • Students will undertake the assessed elements of the project tasks under supervised conditions. • The assessment will take place over multiple sessions up to a combined duration of 14.5 hours. • The project outcomes will consist of a portfolio of evidence submitted electronically. • Students will undertake a project in response to a realistic contextual challenge. • The project is validated by an employer panel, taking into account the client's requirements and the user experience. • The project will consist of planning documentation, an annotated digital portfolio, a prototype digital product, testing evidence and evaluation. • The project will be set and marked by Pearson.

Employer Set Project Assessment Objectives

Assessment Objective			Proportion
AO1	**1. Planning**	Plan an approach to developing solutions to solve problems in response to a brief.	17%
AO2	**2. Application**	Apply knowledge and skills to develop software, create an artefact, fix defects and mitigate risks to security.	43%
AO3	**3. Selecting relevant techniques and resources**	Select relevant tools, techniques and resources to respond to a brief and work in a collaborative environment.	5%
AO4	**4a. Maths skills**	Use appropriate maths skills to realise a project outcome in response to a brief.	3%
	4b. English skills	Use appropriate English skills to communicate technical information to both technical and non-technical audiences.	
	4c. Digital skills	Use appropriate digital skills to realise a project outcome in response to a brief and communicate technical information to both technical and non-technical audiences.	
AO5	**5a. Project Outcome**	Realise a project outcome by producing software and artefacts in response to a brief.	23%
	5b. Review	Review how well digital solutions meet a brief, using reflective evaluation.	9%

5. Occupational Specialist content - Digital Production, Design and Development

Content summary

The Occupational Specialist content covers the knowledge and skills needed to achieve threshold competence across the following areas:

1. Be able to analyse a problem to define requirements and acceptance criteria aligned to user needs

What students need to learn	
1.1 Understand the stages of the software development life cycle and be able to apply them to digital projects	

Life cycle stage: Research and familiarisation
• Explore and understand the initial client request/project brief.
• Carry out research relating to the specific context and market environment, including:
  ○ common problems and risks
  ○ current uses of hardware and software within the identified context
  ○ newly emerging technologies
  ○ existing or potential solutions and how these meet different user needs
  ○ industry/situational-specific guidelines and regulations.
• Identify shortfalls in own skills and knowledge and plan learning opportunities to make up for these shortfalls.
Life cycle stage: Planning and requirement analysis
• Identify business requirements such as new software and amending/increasing security.
• Assess the measurable value of the proposed solution in relation to:
  ○ the user
  ○ the client/business.
• Apply computational thinking (decomposition, pattern recognition and abstraction) to split the problem into discrete objects.
• Define the functional and non-functional requirements of the solution.
• Define the key performance indicators (KPIs) of the solution.
• Identify the performance constraints in digital projects.
• Create user acceptance criteria.
• Schedule projects (tasks, subtasks, milestones).
• Allocate appropriate resources to digital projects.
	E1 E2 E3 E4 E5
	M2 M6 M7 M8 M9 M10
	D1 D2 D3 D4

What students need to learn	
1.1 Understand the stages of the software development life cycle and be able to apply them to digital projects *continued*	

• Estimate costs of digital projects.
• Choose programming language(s) for digital projects based on key criteria, including:
  ○ suitability for the proposed task
  ○ organisational policy
  ○ scalability
  ○ security
  ○ availability of trained staff
  ○ costs
  ○ reliability.
  ○ Identify risks and explore ways to mitigate these risks.
Life cycle stage: Performing a user analysis
• Select and use business analysis models to aid problem solving, including:
  ○ user stories
  ○ activity diagrams
  ○ mind maps
  ○ product road maps
  ○ process diagrams
  ○ entity relationship diagrams.
Life cycle stage: Designing the product
• Create interface designs.
• Plan how to solve key problems (design algorithms).
• Create data requirement designs.
• Create a proof of concept.
• Produce initial testing schedule.
Life cycle stage: Developing and testing the product
• Create a prototype.
• Plan and implement appropriate testing, including:
  ○ module testing
  ○ system integration testing
  ○ automated testing
  ○ user testing and feedback.
Life cycle stage: Deploying/implementing the product
• Install and configure the product.
• Update the product.
Life cycle stage: Maintenance
• Provide system support.
• Provide user support.
• Carry out bug fixing.
• Arrange/carry out user training.	

What students need to learn	
1.1 Understand the stages of the software development life cycle and be able to apply them to digital projects *continued*	
• Release updates to enhance the product. • Understand the value to the organisation of the digital product and the role the software development life cycle plays.	
1.2 Understand the roles and responsibilities of the digital team within the software development life cycle	

• Product owner/client – sets and communicates the requirements of the product.
• Scrum master – facilitates the team to maintain team cohesion, ensuring the team has access to resources they require.
• Technical lead – provides the technical guidance and support for the project team.
• Project manager – plans, organises and manages (budget, scope, schedule, risk and quality) on all phases of a project.
• Software development team:
  ○ systems analyst – analyses the current system and provides the requirements for the new system; uses the requirements to design the new software solution.
  ○ UX/UI designer – interviews users, researches market data and gathers findings to design the user interface.
  ○ software developer/engineer – uses the designs to create and maintain a working solution that is usable, secure and stable.
  ○ operations engineer – ensures the stability of the product.
  ○ security engineer – ensures the security of the product.
• Software testers – responsible for the quality assurance of the software and development.
	E4 E5

1.3 Understand the key features of, and be able to select, appropriate project methodologies when developing a software solution	

Key features of Agile:
• Incremental delivery model:
  ○ sprint
  ○ epic
  ○ story
  ○ spikes.
• Emphasis on producing high-quality products, with initially limited functionality.
• Each increment delivers additional functionality.
• Requirements can be continually altered throughout the project.
• Can lack formal documentation.
	E5
	D1 D3 D6

What students need to learn	
1.3 Understand the key features of, and be able to select, appropriate project methodologies when developing a software solution *continued*	

• Users/clients see working products at each iteration.
• Cost-effective method to get an initial product to market.
• Cancelled/partially completed projects will still result in some usable code/product(s).
Key features of Scaled Agile:
• Expanded incremental delivery model:
  ○ sprint
  ○ epic
  ○ story
  ○ spikes
  ○ product increments.
• Emphasis on producing high-quality products with initially limited functionality.
• Each increment delivers additional functionality.
• Requirements can be continually altered throughout the project.
• Provides cohesive reporting.
• Users/clients see working products at each iteration.
• Cost-effective method to get an initial product to market.
• Cancelled/partially completed projects will still result in some usable code/product(s).
• Final iteration focuses on stability.
Key features of Waterfall:
• Rigidly structured with systematic steps.
• Progress measured through the number of artefacts completed.
• High initial costs with slower return on investment.
• Products cancelled during early stages may result in no usable code/product(s).
• Limited user/client interaction.
• High importance placed on documentation.
Key features of Rapid Application Development (RAD)
• Focus on quick creation of prototypes, which are systematically improved.
• Emphasis on developing all features first and quality second.
• Users/clients may see only partially working products or mock-ups during early iterations.
• Relies heavily on reuse of previously developed code.
• Suitable for small- to medium-scale projects.	

What students need to learn	
1.3 Understand the key features of, and be able to select, appropriate project methodologies when developing a software solution *continued*	

Key features of LEAN:
• Emphasis on reducing 'waste' in software (and projects), which may include:
  o unnecessary or incorrect features
  o repeated tasks
  o unnecessarily complex solutions
  o ineffective communication
  o unnecessary changes.
• Decisions are made last minute to reduce chance of waste.
• Short iteration cycles with fast delivery of working versions.
• Emphasis is placed on producing iterations that are fit for use, and not specifically in delivering all features or requirements.
• Suitable for projects with small teams and when resources are limited.
• Understand features and approaches of user-centred design (UCD) and how it is used within a software development life cycle:
  o considerations for UCD:
  – Who is the user?
  – What does the user want to achieve by using the product?
  – How does the user interact with the product?
  – When does the user interact with the product?
  – Why is the product being used?
  – What is the user experience?
  o characteristics of UCD (empathetic, iterative, interdisciplinary).
  o stages of the UCD iterative approach:
  – Understand the context of use.
  – Specify the user requirements.
  – Design the solution.
  – Assess against requirements.	

What students need to learn	
1.4 Understand and define the functional and non-functional requirements of a software solution	

• Understand the concept of 'secure by design' and how this influences the decisions made regarding functional and non-functional requirements.
• Understand and define functional requirements of a software solution:
  o the inputs required
  o the data needed
  o the data processing that must take place
  o the logic of the system
  o the deployment and usage platforms for the software.
• Understand and define non-functional requirements of a software solution in terms of:
  o security considerations
  o required accessibility features
  o scalability requirements
  o key performance indicators and metrics in relation to:
  – responsiveness
  – load handling
  – reliability
  o user acceptance criteria.
• Understand the use of 'spike testing' as an early product-testing method to establish requirements and determine the extent of the problem and the required scope of a software solution.
	E1 E5
	M2 M6 M7 M8
	D2 D3 D6

1.5 Investigate the current and potential uses of emerging technologies and how they impact on industries	
Investigate the impact on software development of emerging technologies such as: • operational technology (OT) • artificial intelligence (AI) and virtual intelligence (VI) • conversational AI • the Internet of Things (IoT) • machine learning • object recognition • biometrics • computer vision • robotics • cloud services and platforms • blockchain • data lakes and data warehousing • drones • 3D printing • 5G networks.	**E1 E2 E3 E4**

What students need to learn	
1.6 Identify and be able to address personal training needs that can boost job performance during the software development process	

• Ensure the software developer is able to complete the required solution:
  o Identify what further knowledge is needed.
  o Identify what new skills are needed.
  o Determine if the software developer has the ability needed to complete the required solution.
• Use different methods to address the personal training needs required to enable completion of the project:
  o Undertake coaching from a professional or peer.
  o Learn new skills on the job.
  o Carry out self-study.
  o Use online professional forums.
  o Sign up to internet workshops for additional training.
	E5 E6
	D1 D3
	D6

2 Apply ethical principles and manage risks in line with legal and regulatory requirements when developing software

What students need to learn	
2.1 Investigate the legal and regulatory requirements that apply to developing software	
• Investigate and apply legal and regulatory considerations appropriate to the context and market environment in which they are developing software: o intellectual property rights and licenses o consumer protection o age ratings and classifications o advertising laws o data protection and privacy o copyright and patent o gambling legislation o responsibilities concerning staff and employment practices o territorial restrictions o system security o equality and diversity. • Investigate and apply standards for software development: o ISO/IEC/IEEE 90003:2018 o W3C. • Investigate and consider ethical implications that apply to software development: o codes of conduct o professional practice o software licensing o inclusion and diversity.	**E5** **M4 M5 M6** **D1 D5 D6**
2.2 Identify and manage risks that apply to software development	
• Assess the potential risks associated with a developing a software product appropriate to the context and market environment in which it is being developed in terms of: o data and system security (malicious vs accidental damage) o compatibility with other systems o speed of development o meeting functional and non-functional requirements o meeting key performance indicators (KPIs) o legal and ethical considerations o user engagement o product reach o assessment of risk (likelihood vs seriousness) o potential impact of risk o potential ways to mitigate identified risks o contingency planning o ongoing monitoring	**E5** **M4 M5 M6** **D1 D5 D6**

What students need to learn	
2.2 Identify and manage risks that apply to software development continued	

• Investigate policies and procedures that apply to software development to manage and mitigate risks:
  ○ backup
  ○ security
  ○ confidentiality, integrity and availability (CIA)
  ○ personnel, skills and training
  ○ business continuity planning
  ○ disaster recovery planning.
• Be able to make and justify software development decisions and recommendations based on effective assessment of risk vs reward in relation to the context and market environment in which they are developing software.	

3 Discover, evaluate and apply reliable sources of knowledge

What students need to learn	
3.1 Understand and evaluate the reliability of different sources of knowledge	

• Use different sources to find reliable information:
  ○ Use search engines to find reliable websites.
  ○ Read wikis.
  ○ Read blogs.
  ○ Read academic papers.
  ○ Talk to peers.
  ○ Join forums.
  ○ Look at code comments.
  ○ Use code repositories.
• Evaluate the reliability of different sources:
  ○ Reputation – find out who the author is and whether they are credible.
  ○ Understand bias – sources written by a particular individual/organisation.
  ○ Look at the evidence used to support the content of the digital source.
  ○ Cross-referencing/triangulation – compare to other sources.
  ○ Check how current the content is – note the date website was last updated.
	E1 E4 E5 E6 M2 M4 M6 M7 M8 D1 D3 D4 D5 D6

3.2 Select and use techniques to obtain qualitative and quantitative data to be able to evaluate software solutions	
• Use verbal feedback (formal and informal). • Create and deliver surveys/questionnaires. • Select and use performance and use data. • Conduct user observation and complete observation records. • Create a focus group that represents a cross-section of the target audience. • Conduct interviews to gather individual thoughts on a software solution. • Use peer mentoring. • Use formal line management and appraisal procedures.	**E1 E4 E5 E6 M2 M4 M6 M7 M8 D1 D3 D4 D5 D6**

4 Design

What students need to learn	
4.1 Understand the use of common design approaches	

• Be able to use critical thinking in order to select appropriate design approaches when developing a software product, and apply them within a larger project methodology.
• Understand features and approaches of function-orientated (top-down) design:
  ○ how data flows through the system
  ○ systems made up of many sub-systems (functions)
  ○ data flow diagrams to show how each function handles or changes the data
  ○ problems broken down logically (divide and conquer), based on what each function should do within the whole system.
• Understand features and approaches of object-oriented programming (OOP) design:
  ○ core concept of making code reusable through standard OOP structures (methods, classes, objects/instances)
  ○ characteristics of OOP (encapsulation, data abstraction, polymorphism and inheritance)
  ○ common OOP design patterns (creational, structural, behavioural).
• Understand features and approaches of data model design:
  ○ visualisation of the data needed and how it will be organised
  ○ common data models (conceptual, logical, physical, hierarchical relational)
  ○ common data modelling tools:
  – entity–relationship model (ER model)
  – Unified Modeling Language (UML).
• Understand features and approaches of test-driven development (TDD):
  ○ core concept that each new feature starts by writing a test that defines the improvements or function of that feature
  ○ common approaches to test-driven development:
  – add a test
  – run all tests until the new test fails
  – write some code
  – run tests and refactor code
  – repeat.
	E1 E2 E3 E4 E5
	M1 M2 M3 M4 M5 M6 M7 M8 M10
	D1 D2 D4 D6

What students need to learn	
4.1 Understand the use of common design approaches *continued*	

• Understand features and approaches of behaviour-driven development (BDD):
  ○ core concept that development is informed by the required behaviour of a software unit which is specified before coding starts
  ○ desired behaviours must have business value and relate to specified requirements
  ○ common behaviour specification structure (title, narrative, acceptance criteria).
• Understand features and approaches of functional design:
  ○ core concept that a program is built and structured as a set series of modules that perform a single defined process/function
  ○ characteristics of functional programming (recursion, closures, first class functions, higher order functions, anonymous functions, currying)
  ○ common components of functional design (arguments, statements, blocks, procedures, functions/sub-routines).	

4.2 Understand and select platforms used for source code and content management	

• Understand the features of different software development platforms used at different stages of developing a software product:
  coding
  repositories
  branching
  building
  testing
  deployment.
• Understand the differences between proprietary and open source platforms and how these may affect software design.
• Understand the process required to manage the development workflow (Git flow, GitHub flow).
• Understand the differences between different platforms and be able to make informed decisions about which to use and when, in relation to:
  target audience
  budget
  technical features
  staff and training
  ease/speed of development
  platform updates
  security
  reliability
  performance
  compatibility.
	E1 E3
	E5
	M10
	D1 D2
	D3 D4
	D5 D6

What students need to learn	
4.3 Understand and be able to design a software solution	

• Understand and apply user experience (UX) design principles:
  ○ consistency to ensure a product is intuitive for the user, aesthetically pleasing and promotes brand recognition
  ○ information hierarchy to navigate the product more easily
  ○ visual hierarchy to enable more important content to stand out
  ○ confirmation to ensure the user is aware of any actions performed and their impact
  ○ user control to allow navigation of the product, efficient workflow and error correction
  ○ accessibility to ensure the product is appropriate for a wide range of users.
• Understand and apply user interface (UI) design principles:
  ○ wireframes
  ○ style guides
  ○ clickable prototype.
• Understand the features of content management systems and how they are used during design and development:
  ○ search engine optimisation (SEO)
  ○ page/screen management
  ○ social media integration
  ○ analytics tools
  ○ workflow management
  ○ publishing controls
  ○ security features and management
  ○ versioning and rollback
  ○ content repositories
  ○ open APIs
  ○ multilingual support
  ○ support.
• Create program designs using accepted conventions:
  ○ pre-defined code
  ○ flowcharts using standard BCS symbols
  ○ pseudocode using standard notation
  ○ control structures (sequence, selection/branching, iteration)
  ○ data types
  ○ data validation
  ○ data structures.
	E1 E2 E3 E4 E5
	M1 M2 M3 M4 M5 M6 M7 M8 M10
	D1 D2 D3 D4 D6

What students need to learn	
4.3 Understand and be able to design a software solution *continued*	
• Understand considerations when selecting assets (graphics, audio, video, code) to be used in software design, including: o file types o file size o compression o streamed or encoded audio o streamed or embedded video o use of metadata o quality required o bandwidth and storage available o target platform. • Understand the features of different target platforms and how they affect the design and development of a software solution: o operating systems o file systems o server and other infrastructure (physical, virtual) o programming language stack o mobile and web. • Understand how and when to use databases to support software solutions: o user management o e-commerce tasks (stock, order processing, page personalisation) o diagnostics o performance analysis. • Create database designs to support software solutions using: o data dictionary/library o entity relationship diagram (ERD) o normalisation to third normal form. • Identify and understand network integration points: o which data is processed locally, e.g. user input for a computer game o which data is processed remotely, e.g. actions of all players in multi-player environments o how data is transferred between local and remote sources, e.g. remote system calls o how local and remote systems will be connected, e.g. type of network, combination of networks o system boundaries to identify which tasks are carried out locally or remotely o which external systems to integrate with, e.g. OPTA for sports data.	

5 Create solutions in a social and collaborative environment

What students need to learn	
5.1 Understand the reasons for using collaborative techniques	

• Use team working on common projects that allows for:
  ○ a reduction in development time
  ○ better communication
  ○ sharing of knowledge
  ○ development of software development skills
  ○ code reviews
  – paired programming
  – informal walkthroughs
  – formal inspections.
• Understand when to collaborate and when working independently is more appropriate.
	E5
	M10
	D1 D3

5.2 Understand and be able to select appropriate technologies used in a social and collaborative environment	

• Collaborative technologies to aid working as part of a team:
  ○ communication (e.g. email, instant messaging)
  ○ resource management (e.g. cloud storage, back-up, synchronisation)
  ○ knowledge (collaboration hubs, wikis, community forums, news sites)
  ○ documentation for technical and non-technical audiences.
• Code collaboration technologies.
• Version control.
• Source control.
• Integrated Development Environments (IDEs).
	E5
	M10
	D1 D3
	D6

6 Implement a solution using at least two appropriate languages

What students need to learn	
6.1 Select and use languages to create a software solution for a software project appropriate to the context and market environment in which they are developing software	

• Select and use at least two appropriate languages to implement front-end and back-end solutions:
  ○ Python
  ○ C, C# and C++
  ○ Javascript frameworks (Angular, React)
  ○ Java
  ○ Go
  ○ Ruby
  ○ PHP
  ○ SQL
• Node.Js.
• Be able to embed programming languages within HTML5 and CSS as required.
• Be able to select and use appropriate application programming interfaces (APIs), packages, modules and libraries to add functionality and compatibility:
  ○ generating dynamic page content
  ○ containerisation
  ○ stateful vs stateless components
  ○ form handling
  ○ file and data handling:
  – handle local files
  – create, open, read, write, delete and close files on a server
  – send and receive cookies
  – add, delete and modify data in a database
  ○ interface components
  ○ media content
  ○ adaptive/responsive layout
  ○ working with existing applications, operating systems, cloud-based and traditional platforms
  ○ working with specific devices
  ○ communication over a network
  ○ infrastructure as code
  ○ security features:
  – controlling user-access
  – encrypting data.
	E5
	M1 M2 M3 M4 M5 M6 M7 M8 M10
	D1 D2 D3 D4 D6

What students need to learn	
6.1 Select and use languages to create a software solution for a software project appropriate to the context and market environment in which they are developing software *continued*	

• Understand the use of continuous integration pipelines and how they can be used to build and deliver software solutions:
  o the concept of 'continuous integration – continuous deployment' (CI/CD)
  o common stages of continuous integration pipelines:
  – source code control
  – build automation
  – unit test automation
  – deployment automation
  – monitoring.
• Use common coding conventions:
  o naming conventions
  o code annotations/commenting
  o modularisation
  o structure/indentation
  o version control.
• Use good practice when developing digital products (twelve factor principles):
  o One codebase tracked in revision control, many deploys.
  o Explicitly declare and isolate dependencies.
  o Store config in the environment.
  o Treat backing services as attached resources.
  o Strictly separate build and run stages.
  o Execute the app as one or more stateless processes.
  o Export services via port binding.
  o Scale out via the process model.
  o Maximise robustness with fast start-up and graceful shutdown.
  o Keep development, staging and production as similar as possible.
  o Treat logs as event streams.
  o Run admin/management tasks as one-off processes.	

What students need to learn	
6.2 Select and use appropriate tools and features to create user interfaces that apply user experience (UX) design principles	

• Be able to select and use appropriate user interface features:
  ○ images and animation
  ○ audio
  ○ effects
  ○ interactions:
  – user input
  – output/user feedback
  – textual
  – graphical
  – audio
  – haptic
  ○ data visualization:
  – dashboard
  – graphing
  – data presentation.
• Be able to select and use appropriate user interface techniques:
  ○ layout grids
  ○ layout and use of space
  ○ font selection and typesetting
  ○ letter spacing
  ○ line spacing
  ○ justification
  ○ use of colour and contrast
  ○ input focus
  ○ hover controls.
• Be able to make appropriate design decisions, with consideration of:
  ○ browser support
  ○ target device/platform
  ○ user characteristics
  ○ available bandwidth
  ○ style and branding
  ○ accessibility
  ○ user input method, including:
  – voice
  – text
  – touch screen
  – mouse.
	E5
	M1 M4 M7 M8 M10
	D1 D2 D6

What students need to learn	
6.3 Connect code to data sources as part of a software project	

• Create data sources to support software solutions using a database.
• Connect to data sources using different connections:
  ○ application programming interface (API):
  – types of requests and request methods
  – endpoints
  – retrieving and parsing data
  – displaying data
  – API keys
  ○ Java database connectivity (JDBC)
  – core API (application programming interface)
  – driver manager
  – connection statement
  – prepared statement
  – result set
  – SQL queries
  ○ open database connectivity (ODBC)
  – application
  – driver manager
  – driver
  – data source
  ○ connection method
  – database name or data source
  – credentials – username/password
  – optional parameters
  ○ extracting data
  ○ storing data
  ○ updating data
  ○ deleting data
  ○ connecting to network resources, using tools within the development environment.
• Be able to select and use data sources and connection methods appropriate to the context and market environment in which they are developing software.
	E5
	M4 M5 M6 M10
	D1 D3 D4 D5 D6

6.4 Select and use deployment methods for a software project	
Use suitable deployment methods such as: • local installation • network/server installation • mobile platforms • web-based platforms • cloud-based platforms • containerisation • container scheduling platforms.	**E5** **M5 M6 M10** **D1 D4 D6**

7 Test a software solution

What students need to learn	
7.1 Understand, select and apply functional, non-functional and front-end testing	

Be able to select and carry out appropriate functional, non-functional and front-end testing relevant to the component or product being tested and the stage within the software development life cycle.
• Functional testing:
  ○ unit testing
  ○ smoke testing
  ○ integration testing
  ○ system testing.
• Non-functional testing:
  ○ availability testing
  ○ compatibility testing
  ○ configuration testing
  ○ load testing.
• Front-end testing to check:
  ○ code/script performance and functionality
  ○ browser compatibility
  ○ operating system compatibility
  ○ cross-browser performance
  ○ formatting and rendering
  ○ loading times
  ○ responsiveness.
• Security testing:
  ○ vulnerability scanning
  ○ static analysis
  ○ dynamic analysis
  ○ integration analysis.
	E1 E4 E5 E6 M2 M4 M5 M6 M8 M10 D1 D2 D4 D6

7.2 Understand, select and apply testing techniques	
• Acceptance testing. • Alpha testing. • Beta testing. • Black box testing. • White box testing/structural testing.	**E1 E4 E5 E6 M2 M4 M5 M6 M8 M10 D1 D2 D4 D6**

What students need to learn	
7.3 Select appropriate tests and test data to test the functionality of software	

• Purpose of the identified test.
• Test data:
  ○ valid test data
  ○ invalid test data
  ○ valid extreme test data
  ○ invalid extreme test data
  ○ erroneous test data.
• Pre-requisite to each test.
• Expected test results.
• Update the plan to include:
  ○ actual results
  ○ changes made
  ○ retests/regression testing following changes.
	E1 E4 E5 E6
	M2 M4 M5 M6 M8 M10
	D1 D2 D4 D6

8 Change, maintain and support software

What students need to learn	
8.1 Understand the changing nature of digital products and the factors that drive change	

Understand how business-driven development affects the types of maintenance to be performed:
• To prevent identified and foreseeable issues such as:
  ○ changes to regulatory requirements
  ○ compatibility with new products or technology
  ○ changes in business process
  ○ release of a new product/service.
• To correct unforeseen or previously unpreventable errors such as:
  ○ new vulnerabilities due to changes in other products or systems (zero-day)
  ○ targeted attack
  ○ data corruption
  ○ system failures.
• Iterative development of digital products to maintain relevance:
  ○ review following user/client feedback
  ○ developments in technology
  ○ competition with other organisations
  ○ the need to improve efficiency
  ○ the need to future proof products.
	E1 E2 E3 E5
	D3 D6

8.2 Understand the stages involved in the software change management process	

• Identify issues/changes made during the feedback/review process.
• Document developments and changes in a software project.
• Communicate with different audiences:
  ○ technical
  ○ non-technical.
• Plan the changes required.
• Schedule the changes.
• Carry out regression testing.
• Control and release updated products:
  ○ planned
  ○ reactive.
	E1 E2 E3 E5
	M6 M10
	D1 D3 D6

8.3 Understand how to maintain code as part of a larger team.	
Understand and apply good coding principles when developing, adapting and maintaining code: • separating code and using modularity • readability • use of common/accepted coding conventions • informative commenting/annotation within the code • updating change logs and other documentation.	**E1 E4 E5 E6** **M2 M4 M5 M6 M8 M10** **D1 D2 D3 D4 D6**

What students need to learn	
8.4 Understand how to support software users	

• Be able to communicate with technical and non-technical audiences, using appropriate tone and levels of technical vocabulary through:
  ○ face-to-face communication
  ○ remote conferencing
  ○ written communication:
  – blogs
  – formal reports
  – technical documentation
  – release notes
  – user guides/help files
  – FAQs
  ○ visual and audio communication
  – user demonstrations
  – screencast videos
  – narration (recorded voice, text-to-speech)
  ○ machine-readable application programming interface (API) contact.
• Apply systematic processes to users and resolve issues:
  ○ identify or replicate the issue
  ○ investigate the possible cause of issues:
  – user error
  – system error
  – application error
  – security breach.
• Apply testing techniques to:
  ○ identify errors in the system/code
  ○ make changes to the system/code as required
  ○ ensure error does not return
  ○ ensure no additional issues have been caused as a result of the changes made.
• Communicate how and when the issue was resolved to appropriate stakeholders.
• Document lessons learned.
	E1 E4 E5 E6
	M2 M4 M5 M6 M8 M10
	D1 D2 D3 D4 D6

Resources for the delivery of the Occupational Specialist Component content

For the Occupational Specialist Component, the following resources are required:

- IDE and debuggers appropriate to the chosen languages
- automated testing tools for:
  UI testing
  performance testing
  load/stress testing
  compatibility testing
- access to communication, collaboration and data collection tools, including:
  online code repositories (such as GitHub)
  forums (e.g. stack overflow)
  questionnaire/survey tools (e.g. Google forms, SurveyMonkey)
  email
- access to online third-party digital content (e.g. video and graphics)
- screencasting and video editing software
- microphones and headphones.

Scheme of Assessment – Occupational Specialist Component

The *Level 3 T Level Technical Qualification in Digital Production, Design and Development* includes a single Occupational Specialist Component. Therefore, there is a single synoptic assessment for the Occupational Specialist Component, which is an extended 'design, development and implementation' project. The synoptic element of the project is important in order to ensure that students are able to demonstrate threshold competence; this is the principal reason why the occupational specialism is assessed via a single extended project assessment to ensure that students are able to evidence all the skills required by the Performance Outcomes.

The Occupational Specialist Component consists of a number of activities grouped into four substantive tasks.

Each task will be completed during a window set by Pearson, during which you will schedule supervised assessment sessions. In some cases, tasks will also involve opportunities for unsupervised activities, where the requirements of the skills being assessed make this necessary.

Occupational Specialism Project – Digital Production, Design and Development
Externally assessed project: 67 hours **145 marks**
Performance Outcomes In this unit, students will: **PO1:** analyse a problem to define requirements and acceptance criteria, aligned to user needs **PO2:** design, implement and test software **PO3:** change, maintain and support software **PO4:** create solutions in a social and collaborative environment **PO5:** discover, evaluate and apply reliable sources of knowledge **PO6:** apply ethical principles and manage risks in line with legal and regulatory requirements when developing software.
Assessment overview There are four parts to the assessment: **Task 1:** Analysing the problem and designing a solution. **Task 2:** Developing the solution. **Task 3a:** Gathering feedback to inform future development. **Task 3b:** Evaluating feedback to inform further development. Students will respond to a given scenario to complete a substantial Digital Production, Design and development project. Students will be assessed on their application of the skills listed for the Performance Outcomes. The tables below show each assessment area of a typical Occupational Specialist project and the relevant skills that the project will target. Students will not be assessed against specific 'knowledge' outcomes but will be expected to draw on and apply related knowledge to ensure appropriate outcomes when applying the skills in response to an assessment scenario. • Students will undertake the project under a combination of supervised and non-supervised conditions. • The assessment will take place over multiple sessions, up to a combined duration of 67 hours. • The project outcomes will consist of a portfolio of evidence submitted electronically. • Students will respond to a scenario to design and develop a software-based solution. • This project will be set and marked by Pearson.

Performance Outcomes

Performance Outcome		Weighting	
		Raw marks	% of total marks
PO1	Analyse a problem to define requirements and acceptance criteria, aligned to user needs.	26	16.1%
PO2	Design, implement and test software.	69	42.9%
PO3	Change, maintain and support software.	34	21.1%
PO4	Create solutions in a social and collaborative environment.	9	5.6%
PO5	Discover, evaluate and apply reliable sources of knowledge.	13	8.1%
PO6	Apply ethical principles and manage risks in line with legal and regulatory requirements when developing software.	10	6.2%

6. Technical Qualification grading, T Level grading and results reporting

The *T Level Technical Qualification in Digital Production, Design and Development* will be graded and awarded to comply with the requirements of Ofqual's General Conditions of Recognition.

Calculation of the Technical Qualification grade

The Technical Qualification components are awarded at the grade ranges shown in the table below.

Component	Available grade range
Core	A*–E and Unclassified
Occupational Specialist	Unclassified, Pass, Merit, Distinction

The Core Component uses an aggregation of points from each of the three Core Assessments to calculate the A* to E.

Students whose level of achievement for either component is below the minimum judged by Pearson to be of sufficient standard will receive an unclassified U result.

Uniform Mark Scale

Students' raw sub-component marks will be converted to a Uniform Mark Scale (UMS). The UMS is used to convert students' sub-component 'raw' marks into uniform marks. This is done in order to standardise marks from one series to another as assessments may vary in difficulty. For example, a student who achieves the lowest mark worthy of a C grade in the Employer Set Project one series will receive the same uniform mark as a student achieving that same grade and level of performance in another series, regardless of their raw marks.

The maximum number of uniform marks available for each sub-component, and the uniform marks relating to each grade boundary, are fixed. These are shown in the following table:

Grade	Core Exam	Core ESP	Core Overall
Maximum	**240**	**120**	**360**
A*	216 – 240	108 – 120	324 – 360
A	192 – 215	96 – 107	288 – 323
B	168 – 191	84 – 95	252 – 287
C	144 – 167	72 – 83	216 – 251
D	120 – 143	60 – 71	180 – 215
E	96 – 119	48 – 59	144 – 179
U	0 – 95	0 – 47	0 – 143

The Core Examination has two exam papers, the results of which are combined before conversion to UMS.

Calculation of the T Level grade

The *Calculation of qualification grade* table below shows the minimum thresholds for calculating the T Level grade, subject to successful completion of all elements. The table will be kept under review over the lifetime of the T Level.

Calculation of qualification grade			
	Occupational specialism grade		
Core grade	**Distinction**	**Merit**	**Pass**
A*	Distinction*	Distinction	Distinction
A	Distinction	Distinction	Merit
B	Distinction	Merit	Merit
C	Merit	Merit	Pass
D	Merit	Pass	Pass
E	Pass	Pass	Pass

Students who do not meet the minimum requirements for a T Level to be awarded will not be certificated. They may receive a Notification of Performance for individual components.

In order to be awarded the T Level, a student must complete both components and achieve a minimum of a grade E in the Core and a Pass in the Occupational Specialism. In addition, they must successfully complete the other elements of the T Level as required by the Institute for Apprenticeships and Technical Education (the Institute) and the T Level panel, such as 315 hours of industry placement.

Students whose level of achievement for either component is below the minimum judged by Pearson to be of sufficient standard will receive an unclassified U result.

Results reporting

The *T Level Technical Qualification in Digital Production, Design and Development* forms the substantive part of the Digital Production, Design and Development T Level programme. The T Level programme includes other elements that are required to be successfully completed in order for students to be awarded the T Level from the Institute. The Institute will provide T Level certificates to students who successfully complete all elements of the T Level programme.

The Institute will issue T Level results on Level 3 results day in August.

Pearson are not required to issue Technical Qualification certificates to students; instead we will provide component results for assessments that students undertake.

Pearson will issue component results on the results day designated for each assessment window.

7. Entry, delivery and assessment information

Introduction

This section focuses on the key information to deliver the *T Level Technical Qualification in Digital Production, Design and Development*. It is of particular value to programme leaders and examinations officers, who must ensure appropriate arrangements are made for assessments.

Student registration

Shortly after students start their T Level programme, you must make sure they are registered for the Technical Qualification. You will be required to register students as outlined in our Key Dates Schedule, which will be published annually on our T Level webpage.

At the point of registration onto the Technical Qualification, we will ask you to confirm the Occupational Specialist Component(s) the student has chosen to study, or as a minimum provide an indication.

Students can be formally assessed only for a qualification on which they are registered.

If students' intended qualifications change – for example, if a student decides to choose a different occupational specialism – then the Provider must transfer the student appropriately.

Programme delivery

You are free to deliver this Technical Qualification using any form of delivery that meets the needs of your students. We recommend making use of a wide variety of modes, including direct instruction in classrooms or work environments, investigative and practical work, group and peer work, private study and e-learning.

Availability of live assessment

The assessments for the *T Level Technical Qualification in Digital Production, Design and Development* will be scheduled annually as shown in the table below:

Annual Series for Digital Production, Design and Development					
Component	**First assessment**	**Month(s)**	**Window/ set date**	**Exam type**	**Paper/ on-screen**
Core 1	2021	May/June November	Set date and time	Written examination	Paper
Core 2	2021	May/June November	Set date and time	Written examination	Paper
Employer Set Project	2021	May/June November	Window	Task	Paper
Occupational Specialist	2022	February/ May	Task specific: window/set date and time	Task	Paper

In developing an overall plan for delivery and assessment for the qualification, you will need to consider the order in which you deliver the content and when the assessments will take place.

Students must be prepared for external assessment by the time they undertake it. In preparing students for assessment, you will want to take account of required learning time, the relationship with other external assessments and opportunities for retaking.

Language of assessment

Assessment of this qualification will be available in English. All student work must be in English.

Student assessment entry

You must enter students into an assessment window, either for the Core Component or the Occupational Specialist Component, as outlined in our Key dates Schedule.

For the first attempt for the Core Component, you will need to make an entry for **both** the Core examination **and** the Employer Set Project in the same window (i.e. May/June ***or*** November).

For a resit, students can take the Core examination and/or the Employer Set Project in a separate window. Therefore, you will need to make an entry for the window you require the student to resit the assessment in.

For the Occupational Specialist Component, you will need to make an entry for the window the student wishes to sit the assessment in.

Resit arrangements

As per the Ofqual Technical Qualification Handbook, there is no specific resit window permitted. However, students will be able to resit in any assessment window following their first sitting.

Students may resit:

- the Core assessment(s)
- the Employer Set Project
- the assessments for an occupational specialism, or
- any combination of these.

Where a student fails one of the Core assessments, they must resit both assessments and must do so in the same assessment window.

However, where a student has to resit both the Core assessment and the Employer Set Project, they do not need to retake both sets of assessments in the same assessment window. For clarity, where a student resits the Core assessment, he or she is not required to retake the Employer Set Project, and vice versa.

In order to access a resit opportunity, you will need to make an entry for the window you require the student to resit the assessment in; see *Student assessment entry* above. Resits can take place up until two academic years after the end of the final academic year for the cohort within which the relevant student is included.

Access to qualifications and assessments for students with disabilities or specific needs

Assessments need to be administered carefully to ensure that all students are treated fairly, and that results are issued on time to allow students to progress to their chosen progression opportunities.

Equality and fairness are central to our work. Our equality policy requires that all students should have equal opportunity to access our qualifications and assessments, and that our qualifications are awarded in a way that is fair to every student. We are committed to making sure that:

- students with a protected characteristic (as defined by the Equality Act 2010) are not, when they are undertaking one of our qualifications, disadvantaged in comparison to students who do not share that characteristic
- all students achieve the recognition they deserve for undertaking a qualification, and that this achievement can be compared fairly to the achievement of their peers.

For students with disabilities and specific needs, the assessment of their potential to achieve the qualification must identify, where appropriate, the support that will be made available to them during delivery and assessment of the qualification. Please see information below on reasonable adjustments and special consideration.

Further information on access arrangements can be found in the Joint Council for Qualifications (JCQ) document *Access Arrangements, Reasonable Adjustments and Special Consideration for General and Vocational Qualifications*.

Special requirements

Some students may have special needs during their Technical Qualification assessments. In such cases, Providers can apply for special requirements on their behalf.

We have a dedicated webpage for Special Requirements. This includes:

- reasonable adjustments
- access arrangements
- special consideration
- modified formats.

Reasonable adjustments to assessment

The Equality Act 2010 requires an awarding organisation to make reasonable adjustments where a student with a disability would be at a substantial disadvantage in undertaking an assessment.

To ensure students have fair access to demonstrate the requirements of the assessments, a reasonable adjustment is one that is made before a student takes an assessment. In most cases, this can be achieved through a defined time extension or by adjusting the format of evidence. We can advise you if you are uncertain as to whether an adjustment is fair and reasonable. You need to plan for time to make adjustments if necessary.

We have a dedicated webpage for reasonable adjustment where Providers can learn more about the process and apply on behalf of a student.

Reasonable adjustments can help reduce the effects of a disability or difficulty that puts the student at a substantial disadvantage in an assessment, in order to enable them to demonstrate their knowledge, understanding, skills and behaviours to the level of attainment required.

Providers can access the *Application of Reasonable Adjustment for Technical Qualifications* via our dedicated webpage. The Provider's application must be made in line with the policy and a copy of their application must be retained.

For this qualification we do not see anything that might prevent reasonable adjustment providing the student is still able to demonstrate the achievement of the skills being assessed. This is due to the fact that students must achieve threshold competence. As a result there could be some Assessment Objectives and/or Performance Outcomes that must be met as they are mandatory in order to demonstrate threshold competence and meet the requirements for the sector.

Access arrangements

We also have a dedicated webpage for access arrangements, where Providers can learn more about arrangements available for students with special needs. Access arrangements aim to meet the particular needs of an individual student without affecting the integrity of the assessment.

Access arrangements allow students to show what they know and do without changing the integrity or the demands of the assessment, for example by using a reader or scribe. Access arrangements are approved before an examination or assessment and they allow students with special educational needs, disabilities or temporary injuries to access the assessment.

Special needs could include students:

- with known and long-standing learning difficulties
- with physical disabilities (permanent or temporary)
- with sensory impairment
- whose first language is not English
- who have difficulties at or near the time of assessment that may affect their performance in the assessment.

For more information about access arrangements, we suggest Providers refer to the JCQ booklet *Access Arrangements, Reasonable Adjustments*.

Special consideration

A student's assessment performance can sometimes be affected by circumstances out of their control. Special consideration is a post-examination adjustment that compensates students who were suffering from a temporary illness or condition, or who were otherwise disadvantaged at the time of the Technical Qualification assessment.

Exams officers may apply for special consideration on a student's behalf. We have a dedicated webpage for special consideration. This includes an FAQ fact sheet giving Providers answers to any questions or concerns they may have.

Special consideration will adhere to the following:

- There are general guidelines for special consideration in the JCQ booklet *A guide to the special consideration process*. It covers the process that is applied consistently by all Awarding Organisations. AOs will not enter into discussion with students or their parents as to how much special consideration should be applied.
- Special consideration cannot be applied in a cumulative fashion, i.e. on the basis of a domestic crisis at the time of the exam and the student suffering from a viral illness.
- Private students should liaise with the Provider where entries have been made, so that they can apply for special consideration on the private student's behalf.

Dealing with malpractice in assessment

We adhere to the JCQ document *Suspected Malpractice in Examinations and Assessments* in our approach to investigating potential malpractice or breaches of security. These procedures are in line with the Ofqual Conditions of Recognition. All allegations of potential malpractice are investigated, and sanctions imposed where malpractice is proven.

We have two dedicated webpages concerning malpractice:

- The first gives Providers guidance on how to let us know about anything suspicious or incidents of malpractice, in accordance with JCQ regulations.
- The second gives students information about what malpractice is and how to report it.

What does malpractice mean?

Malpractice means acts that undermine the integrity and validity of assessment, the certification of qualifications, and/or that may damage the authority of those responsible for delivering the assessment and certification.

Pearson does not tolerate actions (or attempted actions) of malpractice by students, Provider staff or Providers in connection with Pearson qualifications. Pearson may impose penalties and/or sanctions on students, Provider staff or Providers where incidents (or attempted incidents) of malpractice have been proven.

Malpractice may arise or be suspected in relation to any unit or type of assessment within the qualification. For further details regarding malpractice and advice on preventing malpractice by students, please see Pearson's *Provider guidance: Dealing with malpractice and maladministration in vocational qualifications*, available on our website.

Providers are required to take steps to prevent malpractice and to investigate instances of suspected malpractice. Students must be given information that explains what malpractice is for internal assessment and how suspected incidents will be dealt with by the Provider. The *Provider guidance: Dealing with malpractice and maladministration in vocational qualifications* document gives comprehensive information on the actions we expect you to take.

Pearson may conduct investigations if we believe that a Provider is failing to conduct the assessments according to our policies. The above document gives further information and examples, and details the penalties and sanctions that may be imposed.

In the interests of students and Provider staff, Providers need to respond effectively and openly to all requests relating to an investigation into an incident of suspected malpractice.

Student malpractice

Student malpractice refers to any act by a student that compromises or seeks to compromise the process of assessment, or which undermines the integrity of the qualifications or the validity of results.

Student malpractice in examinations **must** be reported to Pearson using a *JCQ Form M1* (available at www.jcq.org.uk/exams-office/malpractice). The form should be emailed to studentmalpractice@pearson.com. Please provide as much information and supporting documentation as possible. Note that the final decision regarding appropriate sanctions lies with Pearson.

Failure to report malpractice constitutes staff or Provider malpractice.

Tutor/Provider malpractice

Providers are required to inform Pearson's Investigation Team of any incident of suspected malpractice by staff before any investigation is undertaken. Providers are requested to inform the Investigation Team by submitting a JCQ M2(a) form (downloadable from www.jcq.org.uk/malpractice) with supporting documentation to pqsmalpractice@pearson.com.

Where Pearson receives allegations of malpractice from other sources (for example, Pearson staff or anonymous informants), the Investigation Team will conduct the investigation directly or may ask the Provider to assist.

Incidents of maladministration (accidental errors in the delivery of Pearson qualifications that may affect the assessment of students) must also be reported to the Investigation Team using the same method.

Heads of Providers/Principals/Chief Executive Officers or their nominees are required to inform students and Provider staff suspected of malpractice of their responsibilities and rights; see 6.15 of *JCQ Suspected Malpractice in Examinations and Assessments Policies and Procedures*.

In cases of suspected malpractice, Pearson reserves the right to withhold the issuing of results while an investigation is in progress. Depending on the outcome of the investigation, results may be released or withheld.

We reserve the right to withhold results when undertaking investigations, audits and quality assurance processes. You will be notified within a reasonable period of time if this occurs.

Results transfer to Providers

To ensure you are supported, we will communicate with you on and before results day.

Results day will follow the format below:

Assessment Window	Results day
Summer 2021	August 2021 (Level 3 results day)
November 2021	January/February 2022

As we are not required to issue Technical Qualification certificates, T Level certificates or T Level statements of achievement, we will not require you to complete any forms or processes to claim the Technical Qualification from Pearson. Instead, we will issue the results directly to you.

We will make available:

- Scorecards: outlining the achievement in percentage terms against each Assessment Objective
- Results Plus: a service whereby achievement will be presented in an item-by-item format. This means Providers will be able to ascertain trends across and within cohorts, and clearly label the associated Assessment Objective
- Statement of Provisional Results: we will offer a provisional component result slip, clearly watermarked as a provisional component result.

Post-Results Services

Our Technical Qualification post-results services (PRS) and appeals will be implemented in line with Ofqual requirements, paying particular attention to the *Rules and Guidance for Technical Qualifications*, where sections Ofqual TQ13–23 refer to post-results activities, 'Review of Marking or Moderation' (RoMM) or Appeals.

We will provide the following:

- access to student assessment evidence
- appeal
- clerical checks
- expedited review of marking
- review of marking.

Our post-results services webpage will include all the necessary information for you to access the services for the T Level Technical Qualification. There will also be a specific *T Level Technical Qualification Post-Results Service Guide* for Providers to use following our first live assessment series in 2021. This information should be used alongside the *JCQ Post-Results Service Guide*.

PRS will be available after each assessment opportunity. Exams officers will be able to apply for PRS via our online system; however, you must have permission from the student before applying. If a student wishes to apply for PRS they must do so via their Provider. We state within the PRS guidance on our website that we cannot accept appeals directly from students, their parents or other third parties acting on their behalf.

In addition, our unique Results Plus service and a free Access to Scripts service will be available, so that Providers are able to transparently see how marks are awarded.

Appeals process

Our appeals process for the Technical Qualification will reflect industry standards, as outlined by the relevant Ofqual Condition(s) (TQ17–TQ22) relating to appeals.

You will be able to appeal the outcome of marking, decisions made regarding reasonable adjustments or special consideration tariffs applied, and any consequence of malpractice or maladministration investigations by us or other Technical Qualification AOs.

All our investigations will be conducted in accordance with the *JCQ General and Vocational Qualifications Suspected Malpractice in Examinations and Assessments Policies and Procedures*.

8. Provider recognition and approval

Introduction

Our Provider approval process and criteria have been developed in collaboration with other awarding organisations offering Technical Qualifications to ensure you have a seamless experience across awarding organisations when requesting approval to deliver the Technical Qualification.

Approving Eligible Providers as Approved Providers

Eligible Providers, i.e. those who submitted an Intention to Teach and have been approved by the DfE to deliver T Levels, will be required to seek approval from Pearson for each Technical Qualification they wish to deliver. They will do this by completing **one** of the two Provider Application forms:

4. T Level Technical Qualification Delivery Approval Application Form
5. Pearson UK Vocational Provider Approval Application for T Level Technical Qualifications.

The first form is for existing vocational Pearson centres to gain approval to deliver the Technical Qualification.

The second form is for non-vocational Pearson centres, i.e. centres not delivering any other vocational Pearson qualification. These are **new** Providers applying for centre approval to deliver vocational qualifications and the T Level Technical Qualification.

In order to assist **new** Providers in their application process, we have provided additional guidance: *Guidance to assist in the completion of the Pearson Vocational Provider Approval Application for T Level Technical Qualifications*.

In addition, we will also support you in the following ways:

- phone support via our Approvals Team
- on-boarding training via Provider Support.

Provider and Technical Qualification approval

As part of the approval process, you must make sure that the resource requirements listed below are in place before offering the qualification:

- Providers must have appropriate physical resources (for example, equipment, IT, learning materials, teaching rooms) to support the delivery and assessment of the qualification.
- There must be systems in place to ensure continuing professional development for staff delivering the qualification.
- Providers must have in place appropriate health and safety policies relating to the use of equipment by students.
- Providers must deliver the qualification in accordance with current equality and diversity legislation and/or regulations.
- Providers should refer to the Resources for delivery of content section in the components to check for any specific resources required.
- Administration arrangements, including security of live assessments, must be in place.

The methods we use to ensure Providers have the above resources in place include:

- making sure that all Providers complete appropriate declarations at the time of approval
- undertaking approval visits to Providers
- an overarching review and assessment of a Provider's strategy for delivering and quality assuring its technical qualifications.

Providers that do not comply with remedial action plans may have their approval to deliver qualifications removed.

What level of sector knowledge is needed to teach this qualification?

We do not set any requirements for tutors, but recommend that Providers assess the overall skills and knowledge of the teaching team to ensure that they are relevant and up to date. This will give students a rich programme to prepare them for employment in the sector.

For this Technical Qualification, tutors with the following knowledge and skills will be beneficial to the delivery of the programme:

- experience of delivering project-based qualifications
- experience of preparing students for examination-based assessment
- knowledge and understating of Python 3 (core) and at least one other (ideally two) programming language(s)
- knowledge of how to solve problems and develop code to meet specified user/business requirements
- understanding of the role software development plays in organisations
- knowledge of emerging technologies and trends in business, and how software utilises these.

What resources are required to deliver this qualification?

As part of your Provider Approval, you will need to show that the necessary material resources and work spaces are available to deliver this technical qualification. Where specific resources are required to deliver the content, these are stated in the relevant component.

Providers should refer to the *Resources for delivery of content* section in the components to check for any specific resources required.

Quality Assurance for the delivery of the Technical Qualification

All Providers will be subject to the same level of scrutiny for the delivery of the Technical Qualification.

We will monitor you throughout the delivery of the contract. This is to ensure you have appropriate and consistent quality assurance measures in place for the delivery of the Technical Qualification, and to ensure that you maintain ongoing compliance with our quality assurance measures in order to retain your approval status.

To maintain ongoing quality, give support and monitor standards, you will receive a termly phone call, and support visit if necessary, from a Provider Quality Manager (PQM). The PQM will check the quality of delivery, confirm implementation of guide/grade exemplification materials, and confirm you are on track for assessment and are accessing our Provider Support.

We will monitor the following activity that could impact approval status:

- registration patterns
- student outcomes
- quality issues identified by our PQM
- reports of maladministration or malpractice.

The PQM will identify any concerns during the termly phone call, provide support and escalate as required. In order to resolve any issues you may have in meeting our quality assurance measures, the PQM will create an action plan with goals and timescales, and support you through the process. If you do not make adequate progress against the action plan, we will notify the Institute of our intention to remove approval status.

Live assessment monitoring

Each assessment in the Technical Qualification is set and marked by Pearson.

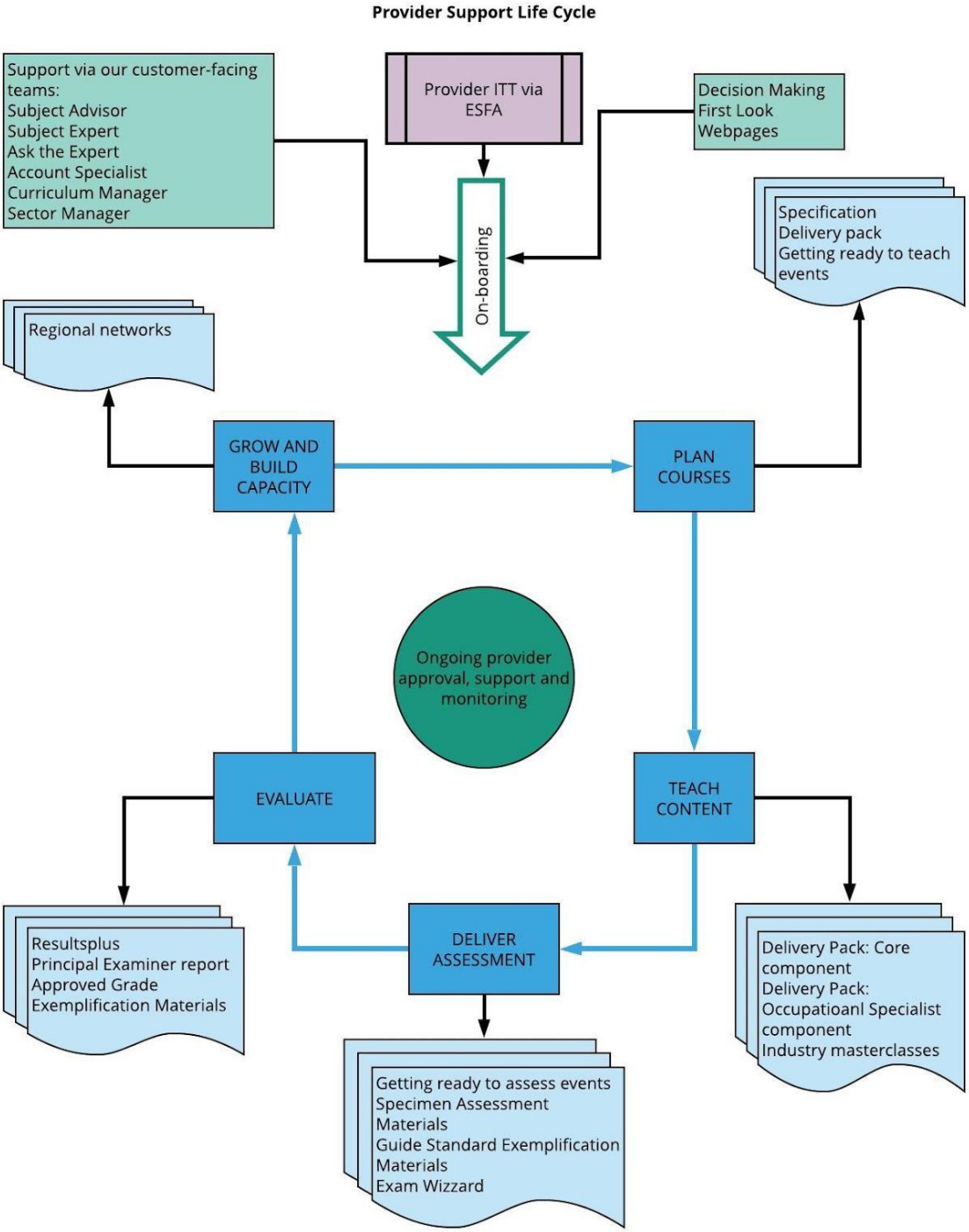
The Core examinations and the Employer Set Project will be sat under exam conditions, following *JCQ's ICE guidance*.

The Occupational Specialist project has different controls depending on the tasks being undertaken by the student. Therefore, full detail of student monitoring will be provided within the assessment materials; these will be published on our T Level webpage before the assessment window commences.

9. Resources and support

The Technical Qualification represents a significant change. It will change how you recruit students, who teaches the occupational specialism, how you incorporate the industry placement, and how you teach and prepare students for external assessment. We will create an innovative range of bespoke support for admin/exams officers, tutors and students that accounts for the step-change the reforms are looking to make in teaching and learning technical skills.

Our aim is to give you a range of support centred on the assessment life cycle, to enable you to deliver the Technical Qualification with confidence. We will include details about our support on our website and inform you via our regular e-bulletins as the materials are launched.



Teaching, learning and assessment materials

1. Plan

- Specification for 2020: content elaboration for the Core and Occupational Specialist Components, guidance and support.
- Delivery Pack: for each component, a Delivery Pack introduction providing advice, help and inspiration.
- Onboarding materials: Provider journey from initial enquiry through to results.

2. Teach

- Delivery Pack: for the Core Component, a number of Topic Guides with lesson activities, industry links and topics linked to assessment.
- Delivery Pack: for the Occupational Specialist Component, an industry project designed with our EVP to enable holistic delivery of the occupational specialism to enhance student experience and prepare for synoptic assessment; used 'off the shelf' or adapted for local or student needs.

3. Assess

- Specimen Assessment Materials (SAMs).
- Guide Standard Exemplification Materials (GSEM).

4. Evaluate

- Principal examiner (PE) report for each component: commentary on performance, including Approved Grade Standard Exemplification Materials.

With the exception of the PE reports, materials will be developed alongside the Technical Qualification. From April 2020 we will have a phased launch of our Provider Support, depending on the resource and where it features within the teaching or assessment life cycle; this is aimed at ensuring you have time to plan, yet aren't bombarded with materials and events within close proximity of each other.

Provider training

Our training will incorporate the above teaching, learning and assessment materials and allow for an in-depth look at content, pedagogy and assessment, providing an opportunity for tutors to network, share ideas and unpick common issues.

The events are targeted at a variety of roles and fit into the Provider journey as follows:

1. Plan

- Admin/Exams Officer: set-up and support on administrative, technical or operational matters.
- First Look: review of the Technical Qualification specification for Providers.
- Getting Ready to Teach: planning, using the support materials, exploring teaching strategies, external and synoptic assessment.

2. Teach

- Regional networks facilitated by our curriculum development managers: sharing good practice and building employer networks to drive innovation and build capacity.
- Industry masterclasses designed and delivered with our EVP: to improve depth of understanding of industry topics.

3. Assess

- Getting Ready to Assess: understanding the standard (using GSEM and mark schemes).

4. Evaluate

- Feedback: examiner feedback and implications for future delivery; demonstrating Exam Wizard and ResultsPlus (see below) to support teaching and exam preparation.

Our events will start in spring 2020 and continue through the contract. To support easy access, the materials will be on our website and events will be delivered in a diverse range of formats:

- face-to-face by a sector specialist
- live online, using interactive technology by a sector specialist
- recorded modules.

Preparing students for external assessment will be new to many tutors. The following two services are unique to Pearson, have proved very popular and will be available for free:

- Exam Wizard is an exam-paper creation tool that allows you to create mock exams and topic tests from a database of sample questions and papers, as well as past papers. Tutors specify the type of assessment they want and a bespoke test with mark scheme and examiner report is created for students to use as practice.
- ResultsPlus is a post-results data analysis tool. It gives item-level analysis by student, class, cohort or cluster of Providers. This allows the user to pinpoint areas of strength and weakness, and to amend teaching and learning to improve student outcomes and motivation.

Provider contact

In addition to the bespoke area of our website which is located here, we will also offer a personal, easy-to-access and expert service via the following:

- Customer Service Account Specialist: named contact for admin/exams officer on administrative or operational matters via phone or email.
- Subject Advisor: named contact for teaching and delivery questions via phone, email, live chat, Facebook, Twitter; provides monthly news promoting support, training and updates.
- Ask the Expert: email address to ask complex or specialist questions relating to content, delivery and assessment.
- Curriculum Development Managers: regional staff who will promote the Technical Qualification, support onboarding and facilitate network events.
- Sector Manager: collect feedback from Providers and employers to identify improvements to our support and services.

Our T Levels Support webpage gives you all the contact details in order to support you.

This includes our:

1. Pearson Support Portal
2. email addresses for administration and teacher support
3. call centre, which is open between 8am and 5pm
4. postal address.

Appendix 1: Glossary of terms used

This is a summary of the key terms used to define the requirements in the components.

Term	Definition
Assess	Give careful consideration to all the factors or events that apply and identify which are the most important or relevant. Make a judgement on the importance of something, and come to a conclusion where needed.
Complete (diagram)	Complete a diagram or process flow that has already been started.
Complete (table)	Provide the missing information for a table/diagram so that it is complete (contains all the necessary information).
Describe	Present two (or more) linked descriptive points on characteristics, features, uses or processes. Do not need to include a justification or reason.
Develop (pseudocode)	Produce a section of code to provide a solution to a problem.
Discuss	Consider the different aspects in detail of an issue, situation, problem or argument, and how they interrelate.
Draw	Produce a diagram, either using a ruler or using freehand OR create a graphical or visual representation of information.
Evaluate	Consider various aspects of a subject's qualities in relation to its context, such as strengths and weaknesses, advantages and disadvantages, pros and cons. Come to a judgment supported by evidence, which will often be in the form of a conclusion.
Explain	Present one point that identifies a reason, way or importance and a second point that justifies/explains the first point. Where used, a third point is a further justification/explanation.
Give	Recall from memory a feature, characteristic or use.
Identify	Select the correct answer from the given context or stimulus.
Label	Correctly indicate parts of a diagram/image/graphical representation.
List	Recall from memory facts, dates, legal implications, etc. More than one.
State	Recall from memory a fact, date, legal implication, etc.

Appendix 2: Pseudocode, flowchart symbols and Python commands

This appendix provides additional information about the digital technologies that students are expected to learn about within the core and occupational specialism content.

This appendix does **not** replace the specification but should be used alongside the specification content to provide additional guidance and scope.

Sections of the specification that do not require additional expansion are not included.

Pseudocode

Data types

STRING

CHARACTER

INTEGER

REAL

FLOAT

BOOLEAN

Type coercion

Type coercion is automatic if indicated by context. For example, 3 + 8.25 = 11.25 (integer + real = real).

Coercion can be made explicit. For example, RECEIVE age FROM (INTEGER) KEYBOARD assumes that the input from the keyboard is interpreted as an INTEGER, not a STRING.

Constants

The value of constants can only ever be set once. They are identified by the keyword CONST.

Two examples of using a constant are shown.

CONST REAL PI

SET PI TO 3.14159

SET circumference TO radius * PI * 2

Data structures

ARRAY

LIST

DICTIONARY

Indices start at zero (0) for all data structures.

When performing 'slicing' operations and other 'sting handling' operations, the data type STRING can be considered a data structure and should be indexed in the same way.

All data structures have an append operator, indicated by &.

Using & with a STRING and a non-STRING will coerce to STRING. For example, SEND 'Fred' & age.

TO DISPLAY, will display a single STRING of 'Fred18'.

Identifiers

Identifiers are sequences of letters, digits and '_', starting with a letter, for example MyValue, myValue, My_Value, Counter2.

Functions

LENGTH()

For data structures consisting of an array or string.

RANDOM(n)

This generates a random number from 0 to n.

Comments

Comments are indicated by the # symbol, followed by any text. A comment can be on a line by itself or at the end of a line.

Devices

Use of KEYBOARD and DISPLAY are suitable for input and output. Additional devices may be required, but their function will be obvious from the context. For example, CARD_READER and MOTOR are two such devices.

Notes

In the pseudocode on the following pages, the < > symbols indicate where expressions or values need to be supplied. The < > symbols are not part of the pseudocode.

Variables and arrays		
Syntax	**Explanation of syntax**	**Example**
SET Variable TO <value>	Assigns a value to a variable.	SET Counter TO 0 SET MyString TO 'Hello world'
SET Variable TO <expression>	Computes the value of an expression and assigns to a variable.	SET Sum TO Score + 10 SET Size to LENGTH (Word)
SET Array [index] TO <value>	Assigns a value to an element of a one-dimensional array.	SET ArrayClass [1] TO 'Ann' SET ArrayMarks [3] TO 56
SET Array TO [<value>, ...]	Initialises a one-dimensional array with a set of values.	SET ArrayValues TO [1, 2, 3, 4, 5]
SET Array [RowIndex, ColumnIndex] TO <value>	Assigns a value to an element of a two-dimensional array.	SET ArrayClassMarks [2,4] TO 92

Note: the same methodology should be used when assigning values in all data structures.

Selection		
Syntax	**Explanation of syntax**	**Example**

IF <expression> THEN
  <command>
END IF
	If <expression> is true then command is executed.	IF Answer = 10 THEN
		  SET Score TO Score + 1
		END IF

IF <expression> THEN
  <command>
ELSE
  <command>
END IF
	If <expression> is true then first
	<command> is executed, otherwise second
	<command> is executed.
		IF Answer = 'correct' THEN
		  SEND 'Well done' TO DISPLAY
		ELSE
		  SEND 'Try again' TO DISPLAY
		END IF

Repetition		
Syntax	**Explanation of syntax**	**Example**

WHILE <condition> DO
  <command>
END WHILE
	Pre-conditioned loop. Executes
	<command> while <condition> is true.
		WHILE Flag = 0 DO
		  SEND 'All well' TO DISPLAY
		END WHILE

REPEAT <command> UNTIL <expression>	Post-conditioned loop. Executes <command> until <condition> is true. The loop must execute at least once.	REPEAT SET Go TO Go + 1 UNTIL Go = 10
REPEAT <expression> TIMES <command> END REPEAT	Count controlled loop. The number of times <command> is executed is determined by the expression.	REPEAT 100-Number TIMES SEND '*' TO DISPLAY END REPEAT

FOR <id> FROM <expression> TO
<expression> DO
  <command>
END FOR
	Count controlled loop. Executes
	<command> a fixed number of times.
		FOR Index FROM 1 TO 10 DO
		  SEND ArrayNumbers [Index] TO DISPLAY
		END FOR

FOR <id> FROM <expression> TO
<expression> STEP <expression> DO
  <command>
END FOR
	Count controlled loop using a step.	FOR Index FROM 1 TO 500 STEP 25 DO
		  SEND Index TO DISPLAY
		END FOR

FOR EACH <id> FROM <expression> DO <command> END FOREACH	Count controlled loop. Executes for each element of an array.	SET WordsArray TO ['The', 'Sky', 'is', 'grey'] SET Sentence to '' FOR EACH Word FROM WordsArray DO SET Sentence TO Sentence & Word & '' END FOREACH

Input/output		
Syntax	**Explanation of syntax**	**Example**
SEND <expression> TO DISPLAY	Sends output to the screen.	SEND 'Have a good day.' TO DISPLAY
RECEIVE <identifier> FROM (type) <device>	Reads input of specified type.	RECEIVE Name FROM (STRING) KEYBOARD RECEIVE LengthOfJourney FROM (INTEGER) CARD_READER RECEIVE YesNo FROM (CHARACTER) CARD_READER

File handling		
Syntax	**Explanation of syntax**	**Example**
READ <File> <record>	Reads in a record from a <file> and assigns to a <variable>. Each READ statement reads a record from the file.	READ MyFile.doc Record
WRITE <File> <record>	Writes a record to a file. Each WRITE statement writes a record to the file.	WRITE MyFile.doc Answer1, Answer2, 'xyz 01'

Subprograms		
Syntax	**Explanation of syntax**	**Example**
PROCEDURE <id> (<parameter>, ...) BEGIN PROCEDURE <command> END PROCEDURE	Defines a procedure.	PROCEDURE CalculateAverage (Mark1, Mark2, Mark3) BEGIN PROCEDURE SET Avg to (Mark1 + Mark2 + Mark3)/3 END PROCEDURE
FUNCTION <id> (<parameter>, ...) BEGIN FUNCTION <command> RETURN <expression> END FUNCTION	Defines a function.	FUNCTION AddMarks (Mark1, Mark2, Mark3) BEGIN FUNCTION SET Total to (Mark1 + Mark2 + Mark3)/3 RETURN Total END FUNCTION
<id> (<parameter>, ...)	Calls a procedure or a function.	Add (FirstMark, SecondMark)

Arithmetic operators	
Symbol	**Description**
+	Add
-	Subtract
/	Divide
*	Multiply
^	Exponent
MOD	Modulo
DIV	Integer division

Relational operators	
Symbol	**Description**
=	equal to
<>	not equal to
>	greater than
>=	greater than or equal to
<	less than
<=	less than or equal to

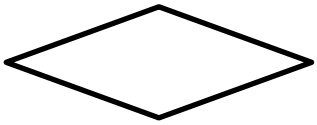
Logical operators	
Symbol	**Description**
AND	Returns true if both conditions are true.
OR	Returns true if any of the conditions are true.
NOT	Reverses the outcome of the expression; true becomes false, false becomes true.

Flowchart symbols

Denotes the start and end of an algorithm

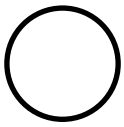
Denotes a process to be carried out

Denotes a sub-process



Denotes a decision to be made

Denotes input or output



Denotes a connection to part of a flowchart that cannot easily be linked using an unbroken flow arrow

Shows the logical flow of the program

Python commands

Handling basic input and output

- input()
- print()
- int()
- str()

Functions and variables

- def
- global

Selection

- if
- else
- elif
- case:

Iteration

- while
- for

Built-in functions and standard library commands

- import
- math

Numerical

- random()
- randint()
- uniform()
- sample()
- range()
- round()
- math.trunc()
- math.floor()
- math.ceil()
- max()
- min()
- count()

String handling

- isupper()
- islower()
- upper()
- lower()
- isalpha()
- split()
- len()

Using data structures (lists and arrays)

- index()
- append()
- insert()
- remove()
- count()
- pop()
- sort()
- in
- not in
- len()

Working with external text files

- open()
- write()
- close()
- read()
- readline()
- readlines()
- line.split()

Additional libraries and commands

For questions in Paper 1 and the Employer Set Project, students will be expected to have a working knowledge of these additional libraries:

- pandas
- Tkinter
- wxPython
- NumPy
- TensorFlow
- Matplotlib.

Explore Pearson's
T Levels offering at
quals.pearson.com/tlevels