

# **Pearson**

# **BTEC Certificates for the Level 3**

# **Software Development**

# **Technician**

# **Apprenticeship Standard**

## **Specification**

BTEC Specialist qualifications

Pearson BTEC Level 3 Certificate in Software Development  
Context and Methodologies

Pearson BTEC Level 3 Certificate in Programming

First teaching May 2019

## **Edexcel, BTEC and LCCI qualifications**

Edexcel, BTEC and LCCI qualifications are awarded by Pearson, the UK's largest awarding body offering academic and vocational qualifications that are globally recognised and benchmarked. For further information, please visit our qualification websites at [www.edexcel.com](http://www.edexcel.com), [www.btec.co.uk](http://www.btec.co.uk) or [www.lcci.org.uk](http://www.lcci.org.uk). Alternatively, you can get in touch with us using the details on our contact us page at [qualifications.pearson.com/contactus](http://qualifications.pearson.com/contactus)

## **About Pearson**

Pearson is the world's leading learning company, with 35,000 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](http://qualifications.pearson.com)

*References to third party material made in this specification are made in good faith. Pearson does not endorse, approve or accept responsibility for the content of materials, which may be subject to change, or any opinions expressed therein. (Material may include textbooks, journals, magazines and other publications and websites.)*

*All information in this specification is correct at time of publication.*

ISBN 978 1446 96022 6

All the material in this publication is copyright  
© Pearson Education Limited 2019

# Contents

<b>1</b>	<b>Introducing BTEC Specialist qualifications</b>	<b>1</b>
	What are BTEC Specialist qualifications?	1
	Sizes of BTEC Specialist qualifications	1
<b>2</b>	<b>Qualifications summary and key information</b>	<b>3</b>
<b>3</b>	<b>Qualification purpose</b>	<b>5</b>
	Qualification objectives	5
	Apprenticeships	5
	Progression opportunities	5
	Industry support and recognition	6
<b>4</b>	<b>Qualification structures</b>	<b>7</b>
	Pearson BTEC Level 3 Certificate in Software Development Context and Methodologies	7
	Pearson BTEC Level 3 Certificate in Programming	7
<b>5</b>	<b>Centre resource requirements</b>	<b>8</b>
	General resource requirements	8
<b>6</b>	<b>Access and recruitment</b>	<b>9</b>
	Prior knowledge, skills and understanding	9
	Access to qualifications for learners with disabilities or specific needs	9
<b>7</b>	<b>Programme delivery</b>	<b>10</b>
<b>8</b>	<b>Assessment</b>	<b>12</b>
	Language of assessment	12
	External assessment	13
	Sample assessment materials	15
	Resits	15
	Administrative arrangements for external assessment	15
	Access arrangements requests	15
	Granting reasonable adjustments	16
	Special consideration requests	16

Conducting external assessments	16
Dealing with malpractice in assessment	17
External assessment	17
Learner malpractice	17
Teacher/centre malpractice	18
Sanctions and appeals	18
<b>9 Centre recognition and approval</b>	<b>20</b>
Approvals agreement	20
<b>10 Quality assurance of centres</b>	<b>21</b>
<b>11 Units</b>	<b>22</b>
Unit 1: Software Development Context and Methodologies	24
Unit 1: Programming	42
<b>12 Further information and useful publications</b>	<b>58</b>
<b>13 Professional development and training</b>	<b>59</b>
<b>14 Suggested teaching resources</b>	<b>60</b>
<b>Annexe A</b>	<b>61</b>
Mapping of the Level 3 Software Development Technician Apprenticeship Standard to the qualification content	61

# 1 Introducing BTEC Specialist qualifications

## What are BTEC Specialist qualifications?

---

BTEC Specialist qualifications are work-related qualifications available from Entry Level to Level 3 in a range of sectors. They give learners the knowledge, understanding and skills they need to prepare for employment in a specific occupational area. The qualifications also provide career development opportunities for those already in work.

BTEC Specialist qualifications put learning into the context of the world of work, giving learners the opportunity to apply their research, skills and knowledge in relevant and realistic work contexts. This applied, practical approach means that learners develop the knowledge, understanding and skills they need for career progression or further study. As such, these qualifications are well suited to support the delivery of the Apprenticeship Standards.

The qualifications may be offered as full-time or part-time courses in schools, colleges, training centres and through employers.

## Sizes of BTEC Specialist qualifications

---

For all regulated qualifications, Pearson specifies a total estimated number of hours that learners will require 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, Pearson identifies the number of Guided Learning Hours (GLH) that we estimate a centre delivering the qualification might provide. Guided learning means activities, such as lessons, tutorials, online instruction, supervised study and giving feedback on performance, that directly involve tutors and assessors in teaching, supervising and invigilating learners. Guided learning includes the time required for learners to complete external assessment under examination or supervised conditions.

In addition to guided learning, other required learning directed by tutors or assessors includes private study, preparation for assessment and undertaking assessment when not under supervision, such as preparatory reading, revision and independent research.

As well as TQT and GLH, qualifications can also have a credit value – equal to one tenth of the TQT, rounded to the nearest whole number.

TQT and credit values are assigned after consultation with the employers and training providers who are delivering the qualifications.

BTEC Specialist qualifications are generally available in the following sizes:

- Award – a qualification with a TQT value of 120 or less (equivalent to a range of 1–12 credits)
- Certificate – a qualification with a TQT value in the range of 121–369 (equivalent to a range of 13–36 credits)
- Diploma – a qualification with a TQT value of 370 or more (equivalent to 37 credits and above).

## 2 Qualifications summary and key information

Qualification title	Pearson BTEC Level 3 Certificate in Software Development Context and Methodologies
Qualification Number (QN)	603/4535/4
Regulation start date	01/05/2019
Operational start date	01/05/2019
Approved age ranges	16–18 19+ Please note that sector-specific requirements or regulations may prevent learners of a particular age from embarking on this qualification. Please see <i>Section 6 Access and recruitment</i> .
Total Qualification Time (TQT)	125 hours
Guided Learning Hours (GLH)	80
Assessment	External assessment – onscreen test.
Grading information	The qualification and units are at a Pass grade.
Entry requirements	No prior knowledge, understanding, skills or qualifications are required before learners register for this qualification. However, centres must follow the Pearson document <i>A guide to recruiting with integrity and enrolling learners onto qualifications</i> (see <i>Section 6 Access and recruitment</i> ).
Funding	The Apprenticeship funding rules can be found at <a href="http://www.gov.uk">www.gov.uk</a>

Qualification title	Pearson BTEC Level 3 Certificate in Programming
Qualification Number (QN)	603/4534/2
Regulation start date	01/05/2019
Operational start date	01/05/2019
Approved age ranges	16–18 19+ Please note that sector-specific requirements or regulations may prevent learners of a particular age from embarking on this qualification. Please see <i>Section 6 Access and recruitment</i> .
Total Qualification Time (TQT)	125 hours
Guided Learning Hours (GLH)	75
Assessment	External assessment – onscreen test
Grading information	The qualification and units are at a Pass grade.
Entry requirements	No prior knowledge, understanding, skills or qualifications are required before learners register for this qualification. However, centres must follow the Pearson document A guide to recruiting with integrity and enrolling learners onto qualifications (see <i>Section 6 Access and recruitment</i> ).
Funding	The Apprenticeship funding rules can be found at <a href="http://www.gov.uk">www.gov.uk</a>

Centres will need to use the Qualification Number (QN) when they seek public funding for their learners. The qualification title, unit titles and QN will appear on each learner’s final certificate. Centres should tell learners this when recruiting them and registering them with Pearson. There is more information about certification in our *UK Information Manual*, available on our website, [qualifications.pearson.com](http://qualifications.pearson.com)



## 3 Qualification purpose

### Qualification objectives

---

This specification covers the following two qualifications:

- Pearson BTEC Level 3 Certificate in Software Development Context and Methodologies
- Pearson BTEC Level 3 Certificate in Programming.

The qualifications develop learners' knowledge and understanding for the job role of software development technician. The qualifications allow learners to achieve the mandatory knowledge modules, in order to complete the on-programme element of the Software Development Technician Apprenticeship.

The qualifications give learners the opportunity to:

- develop knowledge related to software development
- develop technical knowledge and understanding in programming, software context and methodologies
- achieve a Level 3 qualification
- develop personal growth and engagement in learning.

### Apprenticeships

---

The Level 3 Certificate in Software Development, Context and Methodologies and the Level 3 Certificate in Programming are mandatory requirements within the Level 3 Apprenticeship Standard for Software Development Technician. Learners must achieve both qualifications before progressing to the end-point assessment (EPA).

### Progression opportunities

---

Learners who achieve these qualifications, and who have met all other specified requirements of the Apprenticeship Standard, can progress to achieving the full apprenticeship certification that confirms competency in the role of software development technician.

With further training and development, learners can progress to a more senior or complex job role, such as Senior Software Developer. They can also progress to the Level 4 Software Developer Apprenticeship.

## Industry support and recognition

---

The qualification content is based on the requirements set out in the Apprenticeship Standards issued by the Tech Partnership

The qualifications are recognised by:

- Employers: IBM, Capgemini, Microsoft®, Cisco, BT, HP, The Royal Signals, Lowe and Partners, QinetiQ, Weboo, 4Ps Marketing, Fujitsu, Accenture, Atos, CGI, Visa, Contentive, Thales, Ticketmaster®, NCA, Virgin Media.
- Professional organisation/s: BCS Chartered Institute for IT.
- Standards Setting Body: The Tech Partnership.

These qualifications are supported by Digital Skills UK

## 4 Qualification structures

### Pearson BTEC Level 3 Certificate in Software Development Context and Methodologies

---

Learners will need to meet the requirements outlined in the table below before Pearson can award the qualification.

Minimum number of units that must be achieved	1
Minimum number of guided learning hours that must be achieved	80

Unit number	Mandatory unit	Level	Guided learning hours
1	Software Development Context and Methodologies	3	80

### Pearson BTEC Level 3 Certificate in Programming

---

Learners will need to meet the requirements outlined in the table below before Pearson can award the qualification.

Minimum number of units that must be achieved	1
Minimum number of guided learning hours that must be achieved	75

Unit number	Mandatory unit	Level	Guided learning hours
1	Programming	3	75

## 5 Centre resource requirements

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

### General resource requirements

---

- Centres must have appropriate physical resources (for example IT, learning materials, teaching rooms) to support the delivery and assessment of the qualification.
- Staff involved in the assessment process must have relevant expertise and occupational experience. This includes having:
  - current (within the last three years) occupational experience appropriate to the level and breadth of subject areas of the qualifications being assessed
  - substantial knowledge and understanding of the subject areas appropriate to the level, breadth and content of the qualifications. This may be evidenced through having a relevant qualification that is at an equivalent or higher level than the level of the qualification(s) being assessed
  - evidence of continuing professional development (CPD) which may include the achievement of qualifications relevant to the areas being assessed.
- There must be systems in place that ensure continuing professional development (CPD) for staff delivering the qualification.
- Centres must have appropriate health and safety policies in place that relate to the use of equipment by learners.
- Centres must deliver the qualifications in accordance with current equality legislation. For further details on Pearson's commitment to the Equality Act 2010, please see *Section 6 Access and recruitment*. For full details of the Equality Act 2010, visit [www.legislation.gov.uk](http://www.legislation.gov.uk)

## 6 Access and recruitment

Our policy on access to our qualifications is that:

- they should be available to everyone who is capable of reaching the required standards
- they should be free from barriers that restrict access and progression
- there should be equal opportunities for all wishing to access the qualifications.

Centres must ensure that their learner recruitment process is conducted with integrity. This includes ensuring that applicants have appropriate information and advice about the qualification to ensure that it will meet their needs.

Centres should review applicants' prior qualifications and/or experience, considering whether this profile shows that they have the potential to achieve the qualification.

We refer centres to our *Equality, Diversity and Inclusion Policy*, which can be found in the support section of our website.

### Prior knowledge, skills and understanding

---

No prior knowledge, understanding, skills or qualifications are required for learners to register for this qualification.

### Access to qualifications for learners with disabilities or specific needs

---

Equality and fairness are central to our work. Pearson's *Equality, Diversity and Inclusion Policy* document requires all learners to have equal opportunity to access our qualifications and assessments, and that our qualifications are awarded in a way that is fair to every learner.

We are committed to making sure that:

- learners 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 learners who do not share that characteristic
- all learners achieve the recognition they deserve from undertaking a qualification and that this achievement can be compared fairly to the achievement of their peers.

For learners 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 *Section 8 Assessment* for information on reasonable adjustments and special consideration.

## 7 Programme delivery

Centres are free to offer these qualifications using any mode of delivery that meets learners' and employers' needs. It is recommended that centres make use of a wide range of training delivery methods, including direct instruction in classrooms, simulated demonstrations, research or applied projects, e-learning, directed self-study, field visits and role play. Whichever mode of delivery is used, centres must make sure that learners have access to the resources identified in the specification and to the subject specialists delivering the qualifications.

Centres must adhere to the Pearson policies that apply to the different models of delivery. Our *Collaborative and consortium arrangements for the delivery of vocational qualifications policy* document is available on our website.

Those planning the programme should aim to enhance the vocational nature of the qualification by:

- spending time with employers to better understand their organisational requirements and the methods of training that are most suitable, taking into consideration their available resources and working patterns
- collaborating with employers to ensure that learners have opportunities in the workplace to implement the knowledge and skills developed through the training programme
- developing up-to-date and relevant teaching materials that make use of scenarios relevant to the sector and relevant occupation
- giving learners the opportunity to apply their learning in realistic practical activities
- having regular meetings with employers to discuss learner progress, providing feedback and agreeing how any issues will be resolved
- developing projects or assessments with input from employers
- making full use of the variety of experience of work and life that learners bring to the programme
- planning opportunities for the development and practising of skills on the job. On-the-job training presents an excellent opportunity to develop the learner's routine expertise, resourcefulness, craft skills and business-like attitude. It is therefore important that there is intentional structuring of practice and guidance to supplement the learning and development provided through engagement in everyday work activities. Learners need to have structured time to learn and practise their skills separately from their everyday work activities. Teaching and learning methods, such as coaching, mentoring, shadowing, reflective practice, collaboration and consultation, could be used in this structured on-the-job learning.

Where legislation is taught, centres must ensure that it is current and up to date.

Where a qualification is externally assessed, it is essential that learners have covered all of the qualification content before they are tested.

For further information on the delivery and assessment of the new apprenticeships, please refer to the *ESFA funding rules for further education provision*, at:  
[www.gov.uk/government/collections/sfa-funding-rules](http://www.gov.uk/government/collections/sfa-funding-rules)

## 8 Assessment

The table below gives a summary of the assessment methods used for these qualifications.

Qualifications	Assessment method
Level 3 Certificate in Software Development Context methodologies	External assessment (onscreen test)
Level 3 Certificate in Programming	External assessment (onscreen test)

In administering external assessments, centres need to be aware of the specific procedures and policies that apply to, for example, registration, entries and results. More information can be found in our *UK Information Manual*, available on our website.

### Language of assessment

---

External assessments for these qualifications will be available in English only.

A learner taking the qualifications may be assessed in British or Irish Sign Language where it is permitted for the purpose of reasonable adjustment.

Further information on the use of language in qualifications is available in our *Use of languages in qualifications policy* document, available on our website.

For further information on access arrangements, please refer to *Reasonable adjustments to assessments* later in this section.



## External assessment

---

The tables below give information about the type and availability of external assessments that are available for these qualifications. Centres should check this information carefully, together with the relevant qualification specifications and the sample assessment materials, so that they can timetable learning and assessment periods appropriately.

<b>Pearson BTEC Level 3 Certificate in Software Development Context and Methodologies</b>	
Unit title	Software Development Context and Methodologies
Type of assessment	Onscreen multiple choice test
Length of assessment	The external assessment will be 60 minutes.
Number of questions/marks	36
Assessment availability	On demand.
First assessment availability	July 2019

Pearson BTEC Level 3 Certificate in Programming	
Unit title	Programming
Type of assessment	Onscreen multiple choice test.
Length of assessment	The external assessment will be 60 minutes.
Number of questions/marks	36
Assessment availability	On demand
First assessment availability	July 2019

Pearson sets and marks the external assessments.

The external assessment assesses all the learning outcomes in the units to meet the standard specified in the related assessment criteria. All the content in each unit is mandatory for the assessments and will be sampled across different versions of the assessment over time. It is essential, therefore, that learners have full knowledge of the unit content before they are entered for the onscreen test.

Centres need to make sure that learners are:

- fully prepared to sit the external assessments
- entered for the tests at appropriate times, with due regard for resit opportunities as necessary.

Information on registering learners for the test, and the systems requirements for delivering the onscreen tests, is available on our website.

## Sample assessment materials

Each externally-assessed unit has a set of sample assessment materials (SAMs). The SAMs are there to provide an example of what the external assessment will look like in terms of the feel and level of demand of the assessment.

SAMs show the range of possible question types that may appear in the actual assessments and give a good indication of how the assessments will be structured.

While SAMs can be used for practice with learners as with any assessment, the content covered and specific details of the questions asked will change in each assessment.

A copy of each of these assessments can be downloaded from the qualification page on our website.

## Resits

Learners who take the onscreen test and do not perform as expected are allowed the opportunity to resit the assessment. Opportunities for resits are purely at the centre's discretion. Centres will need to ensure that learners are fully prepared against any identified areas of weakness before resitting the assessment.

## Administrative arrangements for external assessment

---

### Access arrangements requests

Access arrangements are agreed with Pearson before an assessment. They allow learners with special educational needs, disabilities or temporary injuries to:

- access the assessment
- show what they know and can do without changing the demands of the assessment.

Access arrangements should always be processed at the time of registration.

Learners will then know what type of arrangements are available and in place for them.

## Granting reasonable adjustments

For external assessment, a reasonable adjustment is one that Pearson agrees to make for an individual learner. A reasonable adjustment is defined for the individual learner and informed by the list of available access arrangements.

Whether an adjustment will be considered reasonable will depend on a number of factors, including:

- the needs of the learner with the disability
- the effectiveness of the adjustment
- the cost of the adjustment; and
- the likely impact of the adjustment on the learner with the disability and other learners.

An adjustment may be judged unreasonable and not approved if it involves unreasonable costs, time frames or affects the integrity of the assessment.

## Special consideration requests

Special consideration is an adjustment made to a learner's mark or grade after an external assessment to reflect temporary injury, illness or other indisposition at the time of the assessment.

An adjustment is made only if the impact on the learner is such that it is reasonably likely to have had a material effect on that learner being able to demonstrate attainment in the assessment.

Centres are required to notify us promptly of any learners who they believe have been adversely affected and request that we give special consideration. Further information can be found in the special requirements section on our website.

## Conducting external assessments

Centres must make arrangements for the secure delivery of external assessments. All centres offering external assessments must comply with the Joint Council for Qualifications (JCQ) document *Instructions for Conducting Examinations*. The current version of this document is available on our website.

## Dealing with malpractice in assessment

---

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

Pearson does not tolerate actual or attempted malpractice by learners, centre staff or centres in connection with Pearson qualifications. Pearson may impose penalties and/or sanctions on learners, centre staff or centres where malpractice or attempted malpractice has been proven.

Malpractice may occur or be suspected in relation to any unit or type of assessment within a qualification. For further details on malpractice and advice on preventing malpractice by learners, please see our *Centre guidance: Dealing with malpractice and maladministration in vocational qualifications* document, available on our website.

The procedures we ask you to adopt vary between units that are internally assessed and those that are externally assessed.

### External assessment

External assessment means all aspects of units that are designated as external in this specification, including preparation for tasks and performance. For these assessments, centres must follow the JCQ procedures set out in the latest version of the Joint Council for Qualifications (JCQ) document *Suspected malpractice in examinations and assessments – Policies and procedures* (available on the JCQ website, [www.jcq.org.uk](http://www.jcq.org.uk)).

In the interests of learners and centre staff, centres need to respond effectively and openly to all requests relating to an investigation into an incident of suspected malpractice.

### Learner malpractice

The head of centre is required to report incidents of suspected learner malpractice that occur during Pearson examinations. We ask centres to complete a JCQ Form M1 (available at [www.jcq.org.uk/exams-office/malpractice](http://www.jcq.org.uk/exams-office/malpractice)) and email it with any accompanying documents (signed statements from the learner, invigilator, copies of evidence, etc.) to the Investigations Processing Team at [candidatemalpractice@pearson.com](mailto:candidatemalpractice@pearson.com)

The responsibility for determining appropriate sanctions or penalties to be imposed on learners lies with Pearson.

Learners must be informed at the earliest opportunity of the specific allegation and the centre's malpractice policy, including the right of appeal. Learners found guilty of malpractice may be disqualified from the qualification for which they have been entered with Pearson.

## Teacher/centre malpractice

The head of centre is required to inform Pearson's Investigations Team of any incident of suspected malpractice by centre staff, before any investigation is undertaken. The head of centre is requested to inform the Investigations Team by submitting a *JCQ Form M2(a)* (available at [www.jcq.org.uk/exams-office/malpractice](http://www.jcq.org.uk/exams-office/malpractice)) with supporting documentation to [pqsmalpractice@pearson.com](mailto:pqsmalpractice@pearson.com). Where Pearson receives allegations of malpractice from other sources (for example Pearson staff, anonymous informants), the Investigations Team will conduct the investigation directly or may ask the head of centre to assist.

Incidents of maladministration (errors in the delivery of Pearson qualifications that may affect the assessment of learners) should also be reported to the Investigations Team using the same method.

Heads of centres/principals/chief executive officers or their nominees are required to inform learners and centre staff suspected of malpractice of their responsibilities and rights, please see section 6.15 of the Joint Council for Qualifications (JCQ) document *Suspected malpractice in examinations and assessments – Policies and procedures*.

Pearson reserves the right in cases of suspected malpractice to withhold the issuing of results/certificates while an investigation is in progress. Depending on the outcome of the investigation, results and/or certificates may not be released or they may be withheld.

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

## Sanctions and appeals

Where malpractice is proven, we may impose sanctions or penalties.

Where learner malpractice is evidenced, penalties may be imposed such as:

- mark reduction for affected external assessments
- disqualification from the qualification
- debarment from registration for Pearson qualifications for a period of time.

If we are concerned about your centre's quality procedures, we may impose sanctions such as:

- working with centres to create an improvement action plan
- requiring staff members to receive further training
- placing temporary blocks on the centre's certificates
- placing temporary blocks on registration of learners
- debarring staff members or the centre from delivering Pearson qualifications
- suspending or withdrawing centre approval status.

The centre will be notified if any of these apply.

Pearson has established procedures for centres that are considering appeals against penalties and sanctions arising from malpractice. Appeals against a decision made by Pearson will normally be accepted only from the head of centre (on behalf of learners and/or members or staff) and from individual members (in respect of a decision taken against them personally). Further information on appeals can be found in our *Enquiries and appeals about Pearson vocational qualifications and end point assessment policy* document, available on our website. In the initial stage of any aspect of malpractice, please notify the Investigations Team (via [pqsmalpractice@pearson.com](mailto:pqsmalpractice@pearson.com)) who will inform you of the next steps.

## 9 Centre recognition and approval

Centres that have not previously offered BTEC Specialist qualifications need to apply for, and be granted, centre recognition as part of the process for approval to offer individual qualifications.

Existing centres will be given 'automatic approval' for a new qualification if they are already approved for a qualification that is being replaced by a new qualification and the conditions for automatic approval are met.

Centres offering mandatory qualifications for the new Apprenticeship Standards must be listed on the Skills Funding Agency's Register of Training Organisations and have a contract to deliver the new Apprenticeship Standards.

Guidance on seeking approval to deliver BTEC qualifications is given on our website.

### Approvals agreement

---

All centres are required to enter into an approval agreement with Pearson, in which the head of centre or principal agrees to meet all the requirements of the qualification specification and to comply with the policies, procedures, codes of practice and regulations of Pearson and relevant regulatory bodies. If centres do not comply with the agreement, this could result in the suspension of certification or withdrawal of centre or qualification approval.



## 10 Quality assurance of centres

Quality assurance is at the heart of vocational qualifications. The centre delivers BTEC Specialist qualifications and will use quality assurance to make sure that their managers and lecturers are standardised and supported. Pearson uses quality assurance to check that all centres are working to national standards. It gives us the opportunity to identify and provide support, if needed, to safeguard certification. It also allows us to recognise and support good practice.

For the qualifications in this specification, the Pearson quality assurance model will follow the process of an annual visit to the centre by a Centre Quality Reviewer, to review centre-wide quality assurance systems.

# 11 Units

Each unit in the specification is set out in a similar way. This section explains how the units are structured. It is important that all tutors, assessors, internal verifiers and other staff responsible for the programme review this section.

Units have the following sections.

## **Unit number**

The number is in a sequence in the specification. Where a specification has more than one qualification, numbers may not be sequential for an individual qualification.

## **Unit title**

This is the formal title of the unit that will appear on the learner's certificate.

## **Level**

All units and qualifications have a level assigned to them. The level assigned is informed by the level descriptors defined by Ofqual, the qualifications regulator.

## **Guided Learning Hours (GLH)**

This indicates the number of hours of activities that directly or immediately involve tutors and assessors in teaching, supervising, and invigilating learners, for example lectures, tutorials, online instruction and supervised study. Units may vary in size.

Pearson has consulted with users of the qualification and has assigned a number of hours to this activity for each unit.

## **Unit introduction**

This is designed with learners in mind. It indicates why the unit is important, what will be learned and how the learning might be applied in the workplace.

## **Learning outcomes**

The learning outcomes of a unit set out what a learner knows, understands or is able to do as the result of a process of learning.

## **Assessment criteria**

The assessment criteria specify the standard the learner is required to meet to achieve a learning outcome.

## **Unit content**

This section sets out the required teaching content of the unit and specifies the knowledge and understanding required for achievement of the unit. It enables centres to design and deliver a programme of learning that will enable learners to achieve each learning outcome and to meet the standard determined by the assessment criteria.

Where it is designed to support apprenticeships, the unit content is informed by the knowledge and understanding requirements of the relevant Apprenticeship Standard.

## **Relationship between unit content and assessment criteria**

All the content in each unit is mandatory for the assessments and will be sampled across different versions of the assessment over time. Learners can be tested on any aspect of the content.

Learners should be asked to complete summative assessment only after the teaching content for the unit or learning outcomes has been covered.

## **Legislation**

Legislation cited in the units is current at time of publication. The most recent legislation should be taught.

## **Essential information for tutors and assessors**

This section gives information to support delivery and the implementation of assessment. It contains the following sub-sections.

- Essential resources – lists any specialist resources needed to deliver the unit. The centre will be asked to make sure that these resources are in place when it seeks approval from Pearson to offer the unit.
- Assessment – this section gives details of the format, structure and any specific conditions of the external assessments.

# Unit 1: Software Development Context and Methodologies

<b>Level:</b>	<b>3</b>
<b>Unit type:</b>	<b>Mandatory for Pearson BTEC Level 3 Certificate in Software Development Context and Methodologies</b>
<b>Assessment type:</b>	<b>External</b>
<b>Guided learning hours:</b>	<b>80</b>

---

## Unit introduction

In this unit, you will gain an understanding of why software is developed and the factors that must be considered to develop high-quality products.

The primary role of a software development technician is to design, create, test and implement code for a variety of platforms. A software development technician helps organisations, in a range of industries, to meet their business needs by developing effective, efficient, robust and often innovative applications. Most industries are increasingly reliant on digital platforms, and understanding why, how and when code for these platforms should be created or updated is a key aspect of this role.

You will gain an understanding of the contexts that drive the creation of software and the factors that influence the design and creation of an application, including technical considerations and how users, customers and the intended platform for delivery all impact on what must be created.

You will also understand common developmental processes that are used in the computing industry, including software development life cycles, the use of data and testing strategies.

Software development technicians will make use of this understanding when producing, or supporting the production of, code as part of a larger team.

## Learning outcomes and assessment criteria

To pass this unit, the learner needs to demonstrate that they can meet all the learning outcomes for the unit. The assessment criteria determine the standard required to achieve the unit.

Learning outcomes		Assessment criteria	
1	Understand the contexts in which software is developed	1.1	Describe the purpose of software that is developed for industries
		1.2	Explain the reasons why the data, and rationale for developing software, varies across different industries
		1.3	Explain why it is important to keep digital processes and systems up to date and responsive to user needs
		1.4	Identify the common digital services, and their features, provided by virtual web-based enterprises
		1.5	Describe how enterprises use web and digital services to engage with customers
		1.6	Explain the software development considerations for different platforms
		1.7	Describe the roles within the software development teams and the tasks they perform

Learning outcomes		Assessment criteria	
2	Understand the methodologies used when developing software	2.1	Identify architecture elements that form the structure of software applications
		2.2	Describe the use of software architecture elements
		2.3	Identify why databases are linked with software applications and the benefits they provide.
		2.4	Identify the reasons for using software configuration management tools during incremental developments of code
		2.5	Explain the benefits of version control when updating code
		2.6	Identify processes that are carried out during different types of testing
		2.7	Describe the phases in the agile software development methodology and how to apply them
		2.8	Describe the phases of the rapid application development (RAD) software development methodology and how to apply them
		2.9	Describe the phases of the waterfall software development methodology and how to apply them
		2.10	Identify the advantages and disadvantages of iterative and linear software development methodologies

## Content

### What needs to be learned

#### Learning outcome 1: Understand the contexts in which software is developed

##### Describe the purpose of software that is developed for industries

- Communication:
  - direct advertising
  - customer/client engagement (websites, social media, mailshots, advertisement campaigns, email)
  - internal communication (email, instant messaging, VoIP).
- Collaboration:
  - cloud technologies
  - content management systems
  - office productivity applications
  - file sharing.
- Customer management:
  - databases
  - customer account control (passwords, usernames, payment systems, preferences)
  - security/data protection (firewalls, encryption, two-factor authentication)
  - customer relationship management (CRM).
- Stock control and order processing:
  - automatic re-ordering
  - item level monitoring
  - 'in-time' ordering
  - sales analysis.
- Manufacturing process control:
  - environmental monitoring (heat, movement, moisture)
  - process/device monitoring (output levels, quality control, machine control, sensor control).
- Design and development:
  - design documentation (flow charts, pseudocode, interface design)
  - rapid prototyping software
  - integrated development environments (code search, debugging, syntax highlighting, visual programming, version control).

## What needs to be learned

### Learning outcome 1: Understand the contexts in which software is developed

#### Explain the reasons why the data, and rationale for developing software, vary across different industries

- Data variation:
  - manufacturing statistics (cost of materials, availability of materials, output, profit, loss, efficiency)
  - test data (test logs, performance metrics, user feedback, white-box testing, black-box testing)
  - customer order history (seasonal trends, stock level monitoring and predictions, loyalty schemes, personalised offers)
  - cookies (page personalisation, direct marketing, user tracking)
  - market research (market trends, user profiles, emerging ideas, growth predictions, marketing).
- Rationale variation:
  - types of user (expert, novice, regular, casual)
  - customer needs (functionality, experience, design, features, reliability)
  - platform on which the software will be used (app, website, closed system)
  - safety considerations (software controlling critical functions, user safety, defensive programming, emerging threats)
  - security of data (legislation, customer confidence, commercial advantage)
  - developments in technology (touch screen and voice control, home integration, digital assistants, Moore's law).

#### Explain why it is important to keep digital processes and systems up to date and responsive to user needs

- Meeting changes in customer requirements (responding to market trends, providing new features, changes in user demographics, improving productivity).
- Customer retention (improved service, responding to competitors, product and service developments).
- Competitive advantage (improved reach, innovation, reduced overheads, greater profits).
- Quality assurance (monitoring, maintaining and improving the desired level of quality in a service or product).



## What needs to be learned

### Learning outcome 1: Understand the contexts in which software is developed

- New technologies:
  - ensuring compatibility of old and new systems
  - avoiding or responding to obsolescence of older hardware and software
  - keeping legacy systems running
  - extending hardware life
  - using old systems for new tasks
  - phased replacement and changeover.

### Identify the common digital services, and their features, provided by virtual web-based enterprises

- Cloud computing services (24/7 access, browser-based applications, common software version, latest version always available, upgrades and maintenance completed by service provider).
- Cloud storage (24/7 access to storage, off-site storage, file sharing, device synchronisation, access from multiple devices).
- Managed backup systems (scalable, recovery procedures, service monitoring off-site storage).
- Social media (public posts, multimedia content, direct and private messaging, direct advertising, polls, analytics).
- Ecommerce sites (product search, real-time stock information, personalisation tool, targeted marketing, secure check out, shopping basket).
- Knowledge sites (wikis, community forums, news sites).

### Describe how enterprises use web and digital services to engage with customers

- Social media (polls, marketing campaigns, customer service representatives).
- Company website (news, updates, social media integration, access to documentation, search engine optimisation).
- Mobile phone app (direct access to a site or service, use of smartphone features to enhance experience, alerts and notifications, widgets).
- Click and collect (free of charge, choice of location, text and email updates when order is ready).
- Web stores (product search, real-time stock information, personalisation tool, targeted marketing, secure check out, shopping basket).
- Online help (FAQs, help pages, live chat, chat bots).

## What needs to be learned

### Learning outcome 1: Understand the contexts in which software is developed

#### Explain the software development considerations for different platforms

- Web:
  - accessible via a browser
  - operating system independent
  - client-server model
  - security considerations (SQL injection, API integration, encryption, cross-site scripting (XSS), error handling)
  - performance considerations (type/version of browser, bandwidth, content optimisation).
- Mobile:
  - type of application (app) to be developed (native, web, hybrid)
  - integration of smartphone features (camera, location, sensors, data connections, touch screen)
  - limitations of mobile hardware (processing power, storage, RAM, battery powered, lack of physical keyboard, mouse)
  - reduced screen space
  - data connections (Wi-Fi, mobile data network, Bluetooth®, NFC)
  - data connection considerations (available bandwidth, power drain, availability of connection, task to be performed).
- Desktop:
  - operating system specific
  - deployment considerations (server, local only, virtual and cloud)
  - user interaction methods (keyboard, mouse, touch screen)
  - hardware and performance considerations (clock speed, number of processing cores, multi-threading, RAM, available storage).
- Cross-platform considerations:
  - user experience (ease of use, familiarity of interface, performance, accessibility, adaptive web pages, device or platform optimisation)
  - branding (house style, social media integration, customer support)
  - page personalisation (cookies, transactional data, search and web history, targeted marketing)
  - accessibility (alt text, screen reader support, adjustable fonts, video narration and subtitles)
  - consistency of experience (similarity of features, access to data, settings, personalisation).

## What needs to be learned

### Learning outcome 1: Understand the contexts in which software is developed

#### Describe the roles of within software development teams and the tasks they perform

- Business analyst:
  - role (investigates, analyses and documents the business processes of an organisation to help define requirements of business)
  - tasks (feasibility study, business requirements analysis, business requirements definition, business case).
- Requirements engineer:
  - role (defines documents and maintains the IT requirements of an organisation based on the identified business requirements)
  - tasks (IT requirements analysis, analyse current trends, liaise with client and other stakeholders, defines the IT requirements of an organisation, defines the goals of a proposed IT system).
- Software designer:
  - role (applies principles of computer science and mathematics to define and plan solutions to problems).
  - tasks (decomposes problems, designs algorithms to solve problems, defines program specifications, produces visual and logical designs).
- Software developer:
  - role (creates program code and associated assets in order to implement the planned solution as a functioning software application)
  - tasks (problem solving, creates program code, integrates systems, software maintenance, updates development logs).
- Software tester:
  - role (plans and implements methodologies to ensure software is functional, secure and meets specified business requirements)
  - tasks (produce test plans, devise tests to ensure all requirements are met, identify test users, collate and interpret feedback, interpret log and performance data).

## What needs to be learned

### Learning outcome 1: Understand the contexts in which software is developed

- Software project manager:
  - role (oversees software development projects, plans and assigns resources to ensure projects meet time, cost and quality requirements)
  - tasks (resource planning, time management, analyse and manage risk, cost benefit analysis, budget control).
- Software release engineer:
  - role (aids quality assurance by working on build of the final package, release and deployment (installation) of the finished application)
  - tasks (deploy/install software, package builds, manage version control, monitor and review installed software, identify and resolve issues with installed software).

## What needs to be learned

### Learning outcome 2: Understand the methodologies used when developing software

#### Identify architecture elements that form the structure of software applications

- User interface (the way in which the user interacts with and receives feedback from the software application).
- Algorithms (a structured, step-by-step solution to a problem).
- Data (values that are transmitted, stored, processed and output by a computer).
- Data structures (the way in which data is organised and stored within an application so that it can be processed).
- Code base (the source code of a software application).
- Code libraries (a set of predefined routines or functions for use in a specific programming language).
- Application programming interfaces (APIs) (a set of definitions that facilitate programming an application to interact with another specific application or service).

#### Describe the use of software architecture elements

- User interface:
  - graphical user interface (GUI)
  - command line interface (CLI)
  - menu-based interface
  - voice recognition
  - audio feedback (text-to-speech, system alerts and notifications)
  - haptics.
- Algorithms:
  - sorting algorithms (bubble sort, quick sort)
  - search algorithms (linear, serial, binary).
- Data:
  - data versus information
  - primitives (character and string, integer, floating point, fixed point, Boolean).

## What needs to be learned

### Learning outcome 2: Understand the methodologies used when developing software

- Data structures:
  - stack
  - queue
  - array
  - list
  - record.
- Code base:
  - distinct and monolithic code bases
  - typical contents of a code base (human written source code, configuration files, property files).
- Code libraries:
  - pre-written code
  - subroutines
  - classes
  - values
  - type specifications.
- Application programming interfaces (APIs):
  - libraries and frameworks
  - operating system APIs
  - remote APIs
  - web APIs.

### Identify why databases are linked with software applications and the benefits they provide

- Why databases are linked to software:
  - Store active program data and user input.
  - User management.
  - Ecommerce tasks (stock, order processing, page personalisation).
  - Diagnostics.
  - Performance analysis.

## What needs to be learned

### **Learning outcome 2: Understand the methodologies used when developing software**

- Benefits of linking databases with software applications:
  - improved access management (permissions, single database for multiple uses)
  - Data can be updated or edited without accessing program code
  - Data can be scaled without affecting the program code
  - Improved data integrity
  - Performance improvements as data is loaded only when needed
  - Ease of maintenance

### **Identify the reasons for using software configuration management tools during incremental developments of code**

- Reduce repetition of tasks.
- Reduce number of unrequired tasks.
- Aids prioritisation of tasks.
- Helps manage resources.
- Helps reduce miscommunication.
- Allows more effective management of simultaneous tasks/updates.
- Allows one person to manage and confirm changes before they are applied to the code base.
- Aids code auditing.
- Facilitates teamwork and collaboration.
- Improves accuracy of defect tracking.

## What needs to be learned

### Learning outcome 2: Understand the methodologies used when developing software

#### Explain the benefits of version control when updating code

- Versioning makes it clearer to developers which file or code base to use.
- Easier to manage development process as changes are made to a single code base.
- Latest file can be easily identified.
- Points in development at which major and minor changes are made can be easily identified.
- Aids documentation of testing and changes.
- Last usable version can be identified if recent version becomes corrupted.
- Editing is non-destructive.
- Easier to identify at what point in a development an error or issue may have occurred.

#### Identify processes that are carried out during different types of testing

- Unit testing:
  - identify processes and input and output requirements
  - identify and isolate code into its smallest testable part
  - plan tests cases
  - identify test data
  - debug code.
- Integration testing:
  - identify units of code that will work together
  - decide on approach to be used (top down, bottom up)
  - define parameters for the way in which the units will work
  - use modules that have been unit tested as the input for the test.
- Performance testing:
  - define performance goals
  - identify suitable metrics
  - deploy manual and automatic testing tools as required
  - analyse data generated by testing tools.



## What needs to be learned

### Learning outcome 2: Understand the methodologies used when developing software

- System testing:
  - test software as a complete package
  - plan destructive testing cases
  - plan non-destructive testing cases
  - compare performance with functional requirements specification.
- Acceptance testing:
  - identify and engage suitable test users
  - deploy users to test the program in real or simulated use scenarios
  - gather feedback from test users
  - compare user feedback against functional and non-functional requirements specification.
- Regression testing:
  - fix errors identified in other stages of testing
  - retest the identified component to check error is fixed
  - retest associated components to ensure no unintentional issues have arisen.
- Load/stress testing:
  - agree acceptable performance parameters (data access speed, load times, number of concurrent users, system availability)
  - identify and deploy browser-level and protocol-level testing
  - expose site to low, normal, high and extreme levels of traffic
  - analyse performance of site against agreed parameters.

### Describe the phases in the agile software development methodology and how to apply them

- Scoping and prioritising projects:
  - project definition
  - duration of project
  - business case
  - feasibility
  - scope of work and resources required.

## What needs to be learned

### Learning outcome 2: Understand the methodologies used when developing software

- Requirements for initial sprint:
  - stakeholder meetings
  - project requirements
  - scheduling and project timeline
  - select team
  - allocate resources
  - define responsibilities.
- Construction and iteration:
  - produce first iteration of product
  - working product containing minimum features
  - client feedback
  - release iteration into production
  - product testing
  - fault fixing
  - product enhancement/development
  - update system
  - user documentation and instructions
  - release current iteration.
- Production and ongoing support:
  - bug fixing
  - user training
  - product enhancement.
- Retirement:
  - identify retirement criteria (redundancy, obsolescence, change of business priority)
  - remove or retire digital product
  - introduce replacement.

### **Describe the phases of the rapid application development (RAD) software development methodology and how to apply them**

- Requirements planning (understanding what needs to be designed and what its function will be):
  - understand problem (problem definition, client requirements)
  - investigate and analyse (document the problem).

## What needs to be learned

### Learning outcome 2: Understand the methodologies used when developing software

- User design, prototyping and construction:
  - the three elements take place at the same time, the process goes around in circles until the product is completed.
- Testing (completed unit tested):
  - debug and test (functional and non-functional testing, interface testing, fixing bugs)
  - documentation (user guide, technical manual).
- Handover to client:
  - product enhancement/development
  - produce/update system and user documentation
  - release current iteration.

### Describe the phases of the waterfall software development methodology and how to apply them

- Requirements specification (understanding what needs to be designed and what its function will be):
  - understand problem (problem definition, client requirements)
  - investigate and analyse (document the problem).
- Development (phase and system design is prepared, specifying system requirements):
  - design (functional and non-functional elements, user interface, design specification)
  - create solution (using coding to create the required solution)
  - implementation (the system is first developed in small programs called units)
  - intergration (all the units developed in the implementation phase are integrated into a system).
- Testing (each unit and the system is tested):
  - debug and test (functional and non-functional testing, interface testing, fixing bugs)
  - documentation (user guide, technical manual).

## What needs to be learned

### Learning outcome 2: Understand the methodologies used when developing software

- Deployment (the product is deployed in the customer environment or released into the market):
  - service management phase (deploy, operate, refine).
- Maintenance (making modifications to the system or an individual component to alter attributes or improve performance):
  - application management (monitor, optimise).

### Identify the advantages and disadvantages of iterative and linear software development methodologies

#### Iterative

- Advantages:
  - adaptive approach that responds to changes
  - allows for direct communication with client
  - increases communication with client
  - increases transparency
  - improved quality by finding and fixing defects or deviations from expectations quickly.
- Disadvantages:
  - focuses on working with software
  - lacks documentation efficiency
  - chances of getting off track as outcomes are not clear.

#### Linear

- Advantages:
  - easy to understand and functional
  - simple to manage as model is rigid
  - allows for more structured testing and analysis.
- Disadvantages:
  - matches only precise needs
  - not applicable for maintenance projects
  - reduces ability to see interim prototypes
  - reduced ability to adapt to change
  - projects can extend beyond original deadlines
  - less suitable for extended or ongoing projects.

## Essential information for tutors and assessors

---

### Essential resources

There are no special resources needed for this qualification.

### Assessment

This unit is externally assessed through an onscreen test, which is set and marked by Pearson. The test lasts for 60 minutes and is worth 36 marks. The assessment is available on demand.

The test assesses all the learning outcomes. The questions in the test are based on each assessment criterion and its associated unit content.

The test consists of multiple-choice items. The questions in the test will not necessarily be sequenced in the order of the criteria in the unit, and they will not rely on or directly follow on from another test item.

A Pass grade is determined by learners achieving a defined cut score for the test.

# Unit 1: Programming

<b>Level:</b>	<b>3</b>
<b>Unit type:</b>	<b>Mandatory for Pearson BTEC Level 3 Certificate in Programming</b>
<b>Assessment type:</b>	<b>External</b>
<b>Guided learning hours:</b>	<b>75</b>

---

## Unit introduction

This unit is about understanding how software is developed and what must be considered when undertaking software development. You will examine the features of effective computer programming and apply computing and programming techniques.

The primary role of the software development technician is to design, create, test and implement code for a variety of platforms. A software development technician helps organisations, in a range of industries, to meet their business needs by developing effective, efficient, robust and often innovative applications. Most industries are increasingly reliant on digital platforms and so understanding why, how and when code for these platforms should be created or updated is a key aspect of this role.

You will gain an understanding of the factors that drive the development of software and the need for project teams within software development.

## Learning outcomes and assessment criteria

To pass this unit, the learner needs to demonstrate that they can meet all the learning outcomes for the unit. The assessment criteria determine the standard required to achieve the unit.

Learning outcomes		Assessment criteria	
1	Understand fundamental principles and concepts of programming languages	1.1	Describe the fundamental principles and concepts within procedural languages
		1.2	Describe the fundamental principles and concepts within object-orientated languages
		1.3	Describe the fundamental principles and concepts within event-driven languages
2	Understand how to implement code	2.1	Interpret variables within programming languages
		2.2	Interpret common programming control structures that are used when developing code
		2.3	Describe what common data structures are used for
		2.4	Describe how standard algorithms are applied within software development
		2.5	Describe how to test and debug programs
3	Understand the need to follow good coding practices	3.1	Identify the elements of common coding standards
		3.2	Explain the reasons for good coding practices
4	Understand database normalisation and how to connect code to specified databases	4.1	Identify reasons for connecting code to specified data sources
		4.2	Describe how to connect code to databases using different connections
		4.3	Describe the benefits and drawbacks of using database normalisation

Learning outcomes		Assessment criteria	
5	Understand the functional and non-functional requirements in coding	5.1	Identify the types of functional and non-functional requirements
		5.2	Describe the tests used for functional and non-functional requirements
6	Understand the principles of good interface design	6.1	Describe the key considerations for good interface design
		6.2	Describe the techniques and technologies used in interface design
7	Know how to build security into software development	7.1	Describe common security threats to software
		7.2	Identify ways to reduce the security threats to software
		7.3	Identify the key techniques for defensive programming
8	Understand how software development integrates into a wider project	8.1	Explain each stage of the software development life cycle
		8.2	Describe the project roles and responsibilities at each stage of software development that integrate into the wider project
		8.3	Explain how effective teamwork contributes to the delivery of software development projects



## Content

### What needs to be learned

#### **Learning outcome 1: Understand fundamental principles and concepts of programming languages**

##### **Describe the fundamental principles and concepts within procedural languages**

- Statements (lines of code in sequence)
- Blocks (one or more declarations or statements)
- Procedures (independent code module that fulfils some concrete task and is referenced within a larger body of source code)
- Modularity (splits program up into small parts or chunks, each one of these chunks contribute to the program and performs a task)
- Tools and techniques (functions, procedures, libraries, parameters, debugging).

##### **Describe the fundamental principles and concepts within object-orientated languages**

- Classes (template for creating object, a user-defined data type, which holds its own data members and member functions)
- Objects (specific instance of a class; it contains real values instead of variables)
- Abstraction (handles complexity by hiding unnecessary details from the user)
- Encapsulation (bundles data and methods that work on that data within one unit)
- Inheritance (reusability, enables new objects to take on the properties of existing objects)
- Polymorphism (the ability of a variable, function or object to take on multiple forms)
- Tools and techniques (predefined functions, templates, integrated development environment (IDE)).

## What needs to be learned

### Learning outcome 1: Understand fundamental principles and concepts of programming languages

#### Describe the fundamental principles and concepts within event-driven languages

- Events (user actions, sensor outputs, or messages from other programs or threads)
- Event handlers (series of subroutines or methods)
- Event loops (main loop in a program that typically waits for the user to trigger something)
- Time driven (code that runs on a time trigger)
- Trigger functions (chooses which event handler needs to be run for the event that has occurred)
- Tools and techniques (use of tool boxes and controls, selection, loops, event handlers, objects and object properties, menus).

## What needs to be learned

### Learning outcome 2: Understand how to implement code

#### Interpret variables within programming languages

- Integer.
- Float.
- Boolean.
- String.
- Character.

#### Interpret common programming control structures that are used when developing code

- Sequence:
  - line-by-line execution.
- Selection:
  - if...then...else
  - ELSEIF (ELIF)
  - nested if
  - case/switch.

## What needs to be learned

### Learning outcome 2: Understand how to implement code

- Iteration:
  - repeat
  - for
  - while
  - do-while.

#### **Describe what common data structures are used for**

- Linked lists.
- Arrays.
- Stacks.
- Queues.

#### **Describe how standard algorithms are applied in software development**

- Sorting:
  - bubble
  - quick
  - insertion
  - merge.
- Searching:
  - serial/linear
  - binary.

#### **Describe how to test and debug programs**

- Tools used to debug programs:
  - logical next line
  - step into
  - step out
  - next instruction
  - break point.
- Ordered stages of testing:
  - black box
  - white box
  - alpha
  - beta.

## What needs to be learned

### Learning outcome 3: Understand the need to follow good coding practices

#### Identify the elements of common coding standards

- Open:
  - free to apply own style for code
  - not limited for language constructs
  - can select any programming language.
- Organisational:
  - common style guide for code
  - guide for language constructs
  - pre-determined programming language.

#### Explain the reasons for good coding practices

- Commenting, documentation (to keep a log of events)
- Consistent indentation (shows logical structure).
- Code grouping (grouping data so that the elements in each group share a common attribute).
- Consistent naming (so that variables are easily identifiable):
  - camel case
  - underscores.
- Don't Repeat Yourself (DRY) principle (to reduce the amount of work required to extend and maintain the software in the future)
- Code refactoring (to improve non-functional attributes of the software).

## What needs to be learned

### Learning outcome 4: Understand database normalisation and how to connect code to specified databases

#### Identify reasons for connecting code to specified data sources

- Extracting data.
- Storing data.
- Updating data.
- Deleting data.

#### Describe how to connect code to databases using different connections:

- Java database connectivity (JDBC), for Java language:
  - ordered stages:
    - import packages
    - register the JDBC drivers
    - open connection
    - create a statement object
    - execute a query and return a result set object
    - process the result set
    - close the result set and statement object
    - close the connection.
- Open database connectivity (ODBC), for non-Java language:
  - ordered stages:
    - allocate the handle
    - declare the ODBC version
    - select the data source or driver
    - allocate a connection handle
    - select connection attributes
    - establish a connection
    - driver manager connection pooling
    - close connection.

## What needs to be learned

### Learning outcome 4: Understand database normalisation and how to connect code to specified databases

#### Describe the benefits and drawbacks of using database normalisation

- Benefits:
  - reduced data duplication
  - reduced data redundancy
  - increased data consistency
  - increased data predictive
  - increased data security
  - increased organisation of data
  - increased speed of searching, sorting and creating indexes
  - automatically delete/update all associated records
  - facilitates use of clustered indexes
  - fewer null values
  - data modification anomalies are reduced.
- Drawbacks:
  - increased number of tables
  - more complex to set up queries.

## What needs to be learned

### Learning outcome 5: Understand the functional and non-functional requirements in coding

#### Identify the types of functional and non-functional requirements

- Functional:
  - calculations
  - technical details
  - data manipulation
  - data processing.
- Non-functional:
  - accessibility
  - efficiency
  - reliability
  - scalability
  - maintainability
  - security
  - robustness.

#### Describe the tests used for functional and non-functional requirements

- Functional testing:
  - unit testing
  - smoke testing
  - integration testing
  - system testing.
- Non-functional testing:
  - availability testing
  - compatibility testing
  - configuration testing
  - load testing.

## What needs to be learned

### Learning outcome 6: Understand the principles of good interface design

#### Describe the key considerations for good interface design

- Learnability.
- Memorability.
- Consistency.
- Visibility.
- Constraints.

#### Describe the techniques and technologies used in interface design

- Techniques used in designing effective user interfaces:
  - simple navigation
  - appropriate colours for target audience
  - align fields effectively
  - error handling
  - simple interface.
- Technologies used in user interface:
  - touch screen
  - keyboard
  - voice recognition
  - gesture interface.



## What needs to be learned

### Learning outcome 7: Know how to build security into software development

#### Describe common security threats to software

- Insider threats (malicious and accidental)
- Denial-of-service (DoS) attacks
- SQL injection
- Buffer overflow
- Insecure application programming interfaces (APIS)
- Malware
  - spyware
  - worms
  - trojans
  - viruses
  - adware
  - ransomware.

#### Identify ways to reduce the security threats to software

- Integrate an industry-standard security model.
- Educate personnel on security software.
- Perform architecture reviews and threat modelling.
- Carry out code reviews.
- Perform penetration tests.

#### Identify the key techniques for defensive programming

- Sanitising inputs.
- Protecting routine from bad data.
- Using assertions to document preconditions and post-conditions.
- Standardising exception handling in the code.
- Performing null checks.

## What needs to be learned

### Learning outcome 8: Understand how software development integrates into a wider project

#### Explain each stage of the software development life cycle

- Planning and requirement analysis:
  - resource allocation
  - project scheduling
  - cost estimation
  - programming language choice:
    - organisational policy
    - expandability
    - availability of trained staff
    - costs
    - reliability.
- Defining requirements:
  - questionnaires
  - interviews
  - focus groups
  - observations
  - studying documentation.
- Designing the product:
  - interface design
  - contractors
  - proof of concept
  - design algorithms.
- Developing the product:
  - prototype
  - unit testing
  - system integration.
- Testing the product:
  - logical
  - functional.
- Deploying/implementing the product:
  - installation configuration
  - updates.
- Maintenance:
  - software support
  - user support.

## What needs to be learned

### Learning outcome 8: Understand how software development integrates into a wider project

#### Describe the project roles and responsibilities at each stage of software development that integrate into the wider project

- Client.
- Subject matter expert (SME).
- Project manager (PM).
- Technical lead.
- Software development team:
  - business analyst
  - requirements engineer
  - software designer
  - software developer
  - software tester
  - software release engineer.

#### Explain how effective teamwork contributes to the delivery of software development projects

- Factors that affect team dynamics:
  - positive attitudes (decision making, interpersonal interactions, performance, productivity)
  - recognition for personal effort (engagement, job satisfaction)
  - loyalty to a specific person/group (recognising poor practice, reporting concerns)
  - competition among colleagues (innovation, motivation)
  - professional behaviour (respect of others, commitment to quality, responsibility, accountability, personal appearance)
  - punctuality (managing own time, working within time limits)
  - cooperation (mutual benefit)
  - conflict management (accommodating, avoiding, compromising).
- Communication skills:
  - non-verbal communication (body language, eye contact, hand gestures)
  - active listening
  - clarity
  - concision
  - confidence
  - empathy.

## What needs to be learned

### **Learning outcome 8: Understand how software development integrates into a wider project**

- Importance of teamwork:
  - achieving project objectives
  - camaraderie
  - motivates unity in the workplace
  - offers different perspective and feedback
  - improves efficiency and productivity
  - learning opportunities
  - promotes workplace synergy.

## Further information for tutors and assessors

---

### Essential resources

There are no special resources needed for this qualification.

### Essential information for assessment

This unit is externally assessed through an onscreen test. Pearson will set and mark the test. The test lasts for 60 minutes and is worth 36 marks. The assessment is available on demand.

The test assesses all of the learning outcomes. The questions in the test are based on each assessment criterion and its associated unit content.

The test consists of 36 multiple-choice questions. The questions in the test will not necessarily be sequenced in the order of the criteria in the unit, and they will not rely on or directly follow on from another question. Colour images/diagrams/graphs may be used for the context of the question or for the answer options.

A Pass grade is determined by learners achieving a defined cut score for the test.

## 12 Further information and useful publications

To get in touch with us, visit our 'Contact us' pages:

- Edexcel, BTEC and Pearson Work-based Learning contact details:  
[qualifications.pearson.com/en/support/contact-us.html](https://qualifications.pearson.com/en/support/contact-us.html)

Books, software and online resources for UK schools and colleges:  
[www.pearsonschoolsandfecolleges.co.uk](http://www.pearsonschoolsandfecolleges.co.uk)

Key publications:

- *A guide to recruiting with integrity and enrolling learners onto qualifications* (Pearson)
- *Access arrangements and reasonable adjustments* (Joint Council for Qualifications (JCQ))
- *BTEC Quality Assurance Handbook* (Pearson)
- *Centre Guide to Managing Quality* (Pearson)
- *Collaborative and consortium arrangements for the delivery of vocational qualifications policy* (Pearson)
- *Enquiries and appeals about Pearson vocational qualifications and end point assessment policy* (Pearson)
- *Equality, diversity and inclusion policy* (Pearson)
- *Recognition of Prior Learning, policy and process* (Pearson)
- *Supplementary guidance for reasonable adjustments and special consideration in vocational internally assessed units* (Pearson)
- *Suspected Malpractice in Examinations and Assessments, Policies and Procedures* (JCQ)
- *UK Information Manual* (Pearson)
- *Use of languages in qualifications policy* (Pearson).

All of these publications are available on our website.

Publications on the quality assurance of BTEC qualifications are also available on our website.

Our publications catalogue lists all the material available to support our qualifications. To access the catalogue and order publications, please visit our website.

### **Additional resources**

If you need further learning and teaching materials to support planning and delivery for your learners, there is a wide range of BTEC resources available.

Any publisher can seek endorsement for their resources and, if they are successful, we will list their BTEC resources on our website.

## 13 Professional development and training

Pearson supports UK and international customers with training related to BTEC qualifications. This support is available through a choice of training options offered on our website.

The support we offer focuses on a range of issues, such as:

- planning for the delivery of a new programme
- planning for assessment and grading
- developing effective assignments
- building your team and teamwork skills
- developing learner-centred learning and teaching approaches
- building in effective and efficient quality assurance systems.

The national programme of training we offer is on our website. You can request centre-based training through the website or you can contact one of our advisers in the Training from Pearson UK team via Customer Services to discuss your training needs.

### **BTEC training and support for the lifetime of the qualifications**

**Training and networks:** our training programme ranges from free introductory events through sector-specific opportunities to detailed training on all aspects of delivery, assignments and assessment. We also host some regional network events to allow you to share your experiences, ideas and best practice with other BTEC colleagues in your region.

**Regional support:** our team of Curriculum Development Managers and Curriculum Support Consultants, based around the country, are responsible for providing advice and support in centres. They can help you with planning and curriculum developments.

To get in touch with our dedicated support teams, please visit our website.

### **Your Pearson support team**

Whether you want to talk to a sector specialist, browse online or submit your query for an individual response, there is someone in our Pearson support team to help you whenever – and however – you need:

- **Subject Advisors:** find out more about our subject advisor team – immediate, reliable support from a fellow subject expert
- **Ask the Expert:** submit your question online to our Ask the Expert online service and we will make sure your query is handled by a subject specialist.

Please visit our website at: [qualifications.pearson.com/en/support/contact-us.html](https://qualifications.pearson.com/en/support/contact-us.html)

# 14 Suggested teaching resources

This section lists resource materials that can be used to support the delivery of the qualifications.

## Suggested reading/resources

Abelson H, Sussman G and Sussman J – *Structure and Interpretation of Computer Programs* (MIT Press, 1996) ISBN 9780262510875

Lutz M and Ascher D – *Learning Python*, 2nd edition (O'Reilly Media, 2004) ISBN 9780596002817

Martin R – *The Clean Coder: A Code of Conduct for Professional Programmers* (Prentice Hall, 2011) ISBN 9780137081073

McGrath M – *C Programming in easy steps*, 4th edition (In Easy Steps Limited, 2012) ISBN 9781840785449

Stroustrup B – *The C++ Programming Language*, 4th edition (Addison-Wesley Professional, 2013) ISBN 9780321563842

## Websites

<a href="http://www.computerscience.org/resources/computer-programming-languages/">www.computerscience.org/resources/computer-programming-languages/</a>	Computer programming languages
<a href="http://www.guidetoprogramming.com/joomla153/">www.guidetoprogramming.com/joomla153/</a>	Guide to programming
<a href="http://programmers.stackexchange.com/">http://programmers.stackexchange.com/</a>	Programmers Stack Exchange
<a href="http://www.wiziq.com/tutorials/procedural-programming">www.wiziq.com/tutorials/procedural-programming</a>	Procedural programming tutorials



## Annexe A

---

### Mapping of the Level 3 Software Development Technician Apprenticeship Standard to the qualification content

---

The grid below maps the knowledge, skills and behaviours (KSBs) of the Level 3 Software Development Technician Apprenticeship to the content covered in the Pearson BTEC knowledge modules for the Level 3 Software Development Apprenticeship Standard.

## KEY

# indicates coverage of the knowledge, skills or behaviours in the qualification.

A blank space indicates no coverage of the knowledge, skills or behaviours in the qualification.

BTEC Specialist qualifications	Certificate in Software Development Context and Methodologies	Certificate in Programming
Writes simple code for discrete software components following an appropriate logical approach to agreed standards (whether web, mobile or desktop applications)	#	#
Applies appropriate secure development principles to specific software components at all stages of development	#	#
Applies industry standard approaches for configuration management and version control to manage code during build and release	#	#
Makes simple connections between code and defined data sources as specified	#	#
Functionally tests that the deliverables have been met or not	#	#
Follows basic analysis models such as use cases and process maps		
Supports the software developers at the build and test stages of the software development life cycle	#	#
Follows organisational and industry good coding practices (including for naming, commenting, etc.)	#	#

BTEC Specialist qualifications	Certificate in Software Development Context and Methodologies	Certificate in Programming
Solves logical problems, seeking assistance when required (including appropriate mathematical application) responds to the business environment and business issues related to software development	#	#
Clearly articulates the role and function of software components to a variety of stakeholders (including end users, supervisors, etc.)	#	#
Operates effectively in their own business's, their customers' and the industry's environments		
Develops user interfaces as appropriate to the organisation's development standards and the type of software development being developed	#	#
Apprentices can demonstrate the full range of skills, knowledge and behaviours required to fulfil their job role		
Apprentices can demonstrate how they contribute to the wider business objectives and show an understanding of the wider business environments	#	
Apprentices can demonstrate the ability to use both logical and creative thinking skills when undertaking work tasks, recognising and applying techniques from both	#	#
Apprentices can show that they recognise problems inherent in, or emerging during, work tasks, and can tackle them effectively	#	#

BTEC Specialist qualifications	Certificate in Software Development Context and Methodologies	Certificate in Programming
Apprentices can manage relationships with work colleagues, including those in more senior roles, customers/clients and other stakeholders, internal or external and as appropriate to their roles, so as to gain their confidence, keep them involved and maintain their support for the task/project in hand. Apprentices can establish and maintain productive working relationships, and can use a range of different techniques for doing so		#
Apprentices can communicate effectively with a range of people at work, one to one and in groups, in different situations and using a variety of methods		#
Apprentices can demonstrate various methods of communication, with an understanding of the strengths, weaknesses and limitations of these, the factors that may disrupt it, and the importance of checking other people's understanding		#

**May 2019**

**For information about Pearson Qualifications, including Pearson Edexcel, BTEC and LCCI qualifications visit [qualifications.pearson.com](http://qualifications.pearson.com)**

**Edexcel and BTEC are registered trademarks of Pearson Education Limited**

**Pearson Education Limited. Registered in England and Wales No. 872828  
Registered Office: 80 Strand, London WC2R 0RL.**

**VAT Reg No GB 278 537121**

