

Qualification Specification for the Knowledge Modules that form part of the BCS Level 3 Software Development Technician Apprenticeship

**Level 3 Certificate in Software Development Context and
Methodologies
BCS Level 3 Certificate in Programming**

Version 3.1

February 2019

Contents

| | |
|---|----------|
| 1. About BCS | 3 |
| 2. Equal Opportunities | 3 |
| 3. Introduction to the qualification | 4 |
| 3.1 Qualification summary | 4 |
| 3.2 Purpose of the qualifications | 4 |
| 3.3 Structure of the qualifications | 5 |
| 3.4 Prior learning | 6 |
| 3.5 Learner progression | 6 |

| | |
|---|----------|
| 4. Units | 7 |
| 4.1 Guidance on the qualifications' content | 7 |
| 4.2 Learning outcomes and assessment criteria | 8 |

| | |
|-----------------------------------|-----------|
| 5. Assessment | 22 |
| 5.1 Summary of assessment methods | 22 |
| 5.2 Availability of assessments | 22 |
| 5.3 Grading | 22 |
| 5.4 Externally assessed units | 22 |
| 5.5 Specimen assessment materials | 22 |
| 5.6 Support materials | 22 |
| 5.7 Access to Assessment | 23 |

| | |
|--------------------------|-----------|
| 6. Contact Points | 23 |
|--------------------------|-----------|

1. About BCS

Our mission as BCS, The Chartered Institute for IT, is to enable the information society. We promote wider social and economic progress through the advancement of information technology science and practice. We bring together industry, academics, practitioners and government to share knowledge, promote new thinking, information the design of new curricula, shape public policy and inform the public.

Our vision is to be a world class organisation for IT. Our 70,000-strong membership includes practitioners, businesses, academics and students in the UK and internationally. We deliver a range of professional development tools for practitioners and employees. A leading IT qualification body, we offer a range of widely recognised qualifications.

2. Equal Opportunities

BCS wishes to ensure good practice in the area of Equal Opportunity. Equality of opportunity extends to all aspects for the provision of BCS qualifications.

3. Introduction to the qualification

3.1 Qualification summary

| Qualification Title | QAN | Accreditation Start |
|---|------------|---------------------|
| Level 3 Certificate in Software Development Context & Methodologies | 603/1191/5 | 09/03/17 |
| BCS Level 3 Certificate in Programming | 603/1192/7 | 09/03/17 |

The two knowledge module qualifications listed above have been developed based on the requirements set out in the Standard issued by Tech Partnership and approved by the Government, details of which can be located in the Assessment Plan ([Click Here](#)) and Occupational Brief ([Click Here](#)) documents.

Apprentices must achieve each of the Ofqual-regulated Knowledge Modules as mentioned in the above table.

All BCS qualifications are subject to our quality assurance and validation process. This ensures that new and revised qualifications are fit for purpose. Qualifications are reviewed to ensure the alignment of the qualification with agreed design principles, regulatory requirements and to ensure accuracy and consistency across units and qualifications. Through our quality assurance and validation process, we ensure the qualification, its units and assessments, are fit for purpose and can be delivered efficiently and reasonably by Training Providers.

3.2 Purpose of the qualifications

The qualifications are designed for apprentices enrolled on the Level 3 Software Development Technician Digital IT Apprenticeship, to provide them with the technical knowledge and understanding they require for their role detailed below:

A Software Development Technician typically works as part of a software development team, to build simple software components (whether web, mobile or desktop applications) to be used by other members of the team as part of larger software development projects. They will interpret simple design requirements for discrete components of the project under supervision. The approach will typically include implementing code, which other team members have developed, to produce the required component. The Software Development Technician will also be engaged in testing that the specific component meets its intended functionality

3.3 Structure of the qualifications

This document covers the following qualifications which are used towards the Level 3 Software Development Technician Digital IT Apprenticeship. The qualifications can be taken in any order however it is recommended that they be completed in the following sequence:

1. BCS Level 3 Certificate in Software Development Context & Methodologies
2. BCS Level 3 Certificate in Programming

| Qualification Level 3 Descriptor | |
|---------------------------------------|--|
| Knowledge descriptor (the holder...) | <ul style="list-style-type: none"> • Understands the business context and market environment for software development • Understands the structure of software applications • Understands all stages of the software development lifecycle • Understands the role of configuration management and version control systems and how to apply them • Understands how to test their code (e.g. unit testing) • Recognises that there are different methodologies that can be used for software development • Understands the particular context for the development platform (whether web, mobile, or desktop applications) • Understands their role within their software development team • Understands how to implement code following a logical approach • Understands how their code integrates into the wider project • Understands how to follow a set of functional and non-functional requirements • Understands the end user context for the software development activity • Understands how to connect their code to specified data sources • Understands database normalisation • Understands why there is a need to follow good coding practices • Understands the principles of good interface design. • Understands the importance of building in security to software at the development stage |
| Skills descriptor (the holder can...) | <ul style="list-style-type: none"> • Logical and creative thinking skills • Problem solving skills • Ability to work independently and to take responsibility • Can use own initiative • A thorough and organised approach • Ability to work with a range of internal and external people • Ability to communicate effectively in a variety of situations • Maintains productive, professional and secure working environment |

3.4 Prior learning

The only pre-requisite to take the qualifications is enrolment on the Level 3 Software Development Technician Digital IT Apprenticeship.

Individual employers will set the selection criteria for enrolment onto the Apprenticeship, but this is likely to include five GCSEs, (especially English, Mathematics and a Science or Technology subject); a relevant Level 2 Apprenticeship; other relevant qualifications and experience; or an aptitude test with a focus on IT skills.

3.5 Learner progression

This document covers the qualifications that are part of the Level 3 Software Development Technician apprenticeship. The qualifications must be completed to allow the apprentice to progress onto the End-Point-Assessment, detailed below:

The final, end point assessment is completed in the last few months of the apprenticeship. It is based on

- *a portfolio – produced towards the end of the apprenticeship, containing evidence from real work projects which have been completed during the apprenticeship, usually towards the end, and which, taken together, cover the totality of the standard, and which is assessed as part of the end point assessment*
- *a project - giving the apprentice the opportunity to undertake a business-related project over a one-week period away from the day to day workplace*
- *an employer reference*
- *a structured interview with an assessor - exploring what has been produced in the portfolio and the project as well as looking at how it has been produced*

An independent assessor will assess each element of the end point assessment and will then decide whether to award successful apprentices with a pass, a merit or a distinction.

4. Units

4.1 Guidance on the qualifications' content

The content for each qualification has been developed based on the criteria set out in the Occupational Brief.

| Qualification Title | TQT (Guided Learning + Direct Study + Assessment) |
|---|---|
| Level 3 Certificate in Software Development Context & Methodologies | 125 (83h + 41h + 1h) |
| BCS Level 3 Certificate in Programming | 169.5h (116h + 52.5h + 1h) |

4.2 Learning outcomes and assessment criteria

| Qualification Name | Learning outcomes The learner will.... | Assessment Criteria The learner can... |
|--|---|---|
| Level 3 Certificate in Software Development Context & Methodologies | Understand the business context and market environment for software development. | Understand how similar software development processes and methods are used across a range of industries, but can be based on very different rationale. <ul style="list-style-type: none"> • data. |
| | | Identify the factors that may lead to the development of different information systems within or across a range of industry sectors, including. <ul style="list-style-type: none"> • business requirements; • project timescales; • budgets; • resources and skills availability; • product and project risks. |
| | | <ul style="list-style-type: none"> • Explain why businesses need to keep digital processes up to date and web systems responsive to user needs. |
| | | <ul style="list-style-type: none"> • Explain the difference between virtual web based enterprises and companies that use web and digital services with respect to customer and client engagement. |
| | Recognise that there are different methodologies that can be used for software development. | Identify the main features of sequential development methods and approaches. <ul style="list-style-type: none"> • Waterfall. |
| | | Identify the main features of iterative (incremental) development methods and approaches. <ul style="list-style-type: none"> • Agile. |

| Qualification Name | Learning outcomes The learner will.... | Assessment Criteria The learner can... |
|--------------------|---|--|
| | | Distinguish between the use of different software development methodologies and approaches, considering their suitability and application to the project. <ul style="list-style-type: none"> • Agile; • Waterfall. |
| | Understands the roles within the software development team. | Describe the main roles within software development teams. <ul style="list-style-type: none"> • requirements engineer; • business analyst; • software designer; • software developer; • software tester; • software project manager; • software release engineer. |
| | | Distinguish how the different roles (as listed in 3.1) relate / work with each other and their key accountabilities, in order to complete specific activities and tasks. |
| | | Recognise the key external roles and processes that interface to the roles within the software development team (as listed in 3.1). <ul style="list-style-type: none"> • customers; • end-users; • operation's processes and personnel; • service management processes and personnel. |
| | | Recognise that collaborative approaches are especially important in Agile development and DevOps practices. |

| Qualification Name | Learning outcomes The learner will.... | Assessment Criteria The learner can... |
|--------------------|---|---|
| | Understand the structure of software applications and the particular context for the development platform (whether web, mobile, or desktop applications). | <p>Identify the different components that contribute to the underlying architecture of software applications.</p> <ul style="list-style-type: none"> • code and libraries; • data; • application components; • application interfaces; <ul style="list-style-type: none"> ○ network and hardware platforms • reference to the OSI (Open Systems Interconnection) model. <p>• Describe the use of data sources in software applications for storage and retrieval of information.</p> <p>Explain the features of the following platforms in context of software development, deployment and underlying architecture.</p> <ul style="list-style-type: none"> • web; • desktop; • mobile; • server; • cloud. <p>Distinguish the characteristics of software development that are impacted by the deployment of software.</p> <ul style="list-style-type: none"> • on multiple platforms; • to a single platform. |

| Qualification Name | Learning outcomes The learner will.... | Assessment Criteria The learner can... |
|--------------------|---|---|
| | Understands all stages of the software development lifecycle. | Recognise that there are several ways to represent the terminology and phases of the SDLC. <ul style="list-style-type: none"> • feasibility study; • requirements analysis; • design; • code development; • testing; • deployment / implementation; • maintenance. |
| | | Summarise the phases of the SDLC. |
| | | Identify the main activities of each of the phases of the SDLC in terms of inputs, activities and outputs. |
| | | Recognise the relationship between the phases of the SDLC and the roles within the software development team. |
| | Understand how to test code (e.g. unit testing). | Recognise why testing is necessary, including principles of: <ul style="list-style-type: none"> • early testing; • risk reduction; • conformance to functional and non-functional requirements; • finding and reporting defects; • the difference between testing and debugging. |
| | | Summarise the different levels of testing within the SDLC. <ul style="list-style-type: none"> • unit; • integration; • system; • acceptance. |

| Qualification Name | Learning outcomes The learner will.... | Assessment Criteria The learner can... |
|--------------------|---|--|
| | | <p>Describe how unit testing follows the fundamental test process consisting of:</p> <ul style="list-style-type: none"> • test planning, monitoring and control, including maintaining traceability between requirements and testing artefacts; • test analysis and design; • test implementation and execution; • evaluating exit criteria and reporting. |
| | | <p>Identify the different types and techniques for software testing that are available and why they would be used, including:</p> <ul style="list-style-type: none"> • functional testing; • non-functional testing; <ul style="list-style-type: none"> ○ security ○ performance ○ reliability • reviews and static analysis; • white box testing (structure-based); • black box testing (specification-based). |
| | | <p>Recognise the tool types used to support software testing and their main purpose.</p> <ul style="list-style-type: none"> • test management; • static testing; • test execution; • performance / load / stress testing. |

| Qualification Name | Learning outcomes The learner will.... | Assessment Criteria The learner can... |
|--------------------|--|---|
| | Understand the role of configuration management and version control systems and how to apply them. | <p>Explain how configuration management tools and techniques are used to control and manage the different software development artefacts through the phases of the SDLC and live operation, including:</p> <ul style="list-style-type: none"> • requirements documentation; • code; • test scripts. <p>Summarise the main features and benefits of version control for the development of code including:</p> <ul style="list-style-type: none"> • change history; • concurrent working; • tracking and preventing conflicts; • traceability; • security. <p>Explain how version control can be used for software and software artefacts.</p> <ul style="list-style-type: none"> • that are being developed for use on multiple platforms; • where similar but slightly different versions need to be produced. |

| Qualification Name | Learning outcomes The learner will.... | Assessment Criteria The learner can... |
|--|---|---|
| BCS Level 3 Certificate in Programming | Understand how to implement code, following a logical approach. | Explain the fundamental concepts of programming. <ul style="list-style-type: none"> • procedural vs. object-oriented vs. functional programming; • compiled vs. interpreted. |
| | | Demonstrate the core constructs used when writing code. <ul style="list-style-type: none"> • classes; • objects; • methods; • variables; • logic operators; <ul style="list-style-type: none"> ○ AND ○ OR ○ NOT ○ NAND ○ NOR ○ XOR • control structures. <ul style="list-style-type: none"> ○ iteration ○ selection ○ sequence |
| | | Explain and demonstrate how algorithms are used. <ul style="list-style-type: none"> • encryption; • searching; • sorting. |

| | | |
|--|---|--|
| | | <p>Explain and demonstrate how data structures are used and how data is represented in software code.</p> <ul style="list-style-type: none"> • types of data; <ul style="list-style-type: none"> ○ integer o floating ○ Boolean ○ Character ○ String • variables; • lists, stacks, arrays. |
| | | <p>Describe how to write software code in order to solve problems.</p> <ul style="list-style-type: none"> • describe how programs are structured; <ul style="list-style-type: none"> ○ instructions ○ sub-routines ○ pseudocode ○ data definitions and links ○ comments • describe modularity and the rational re-use of code. <ul style="list-style-type: none"> ○ design patterns ○ library functions ○ frameworks |
| | | <p>Understand the fundamental concept of Test Driven Development (TDD).</p> |
| | <p>Understand how code integrates into the wider project.</p> | <p>Describe the activities undertaken in the following stages of software development:</p> <ul style="list-style-type: none"> • design; • code development; • testing. |
| | | <p>Outline the activities undertaken in the following stages of software development:</p> <ul style="list-style-type: none"> • feasibility study; • requirements analysis; • deployment / implementation. |

| | | |
|--|--|--|
| | | <p>Understand software development activities for the following roles:</p> <ul style="list-style-type: none"> • requirements engineer; • business analyst; • software designer; • software developer; • software tester; • software release engineer. |
| | | <p>Describe the key business concepts and artefacts that must be considered during a software development project.</p> <ul style="list-style-type: none"> • processes and procedures; <ul style="list-style-type: none"> ○ business process management as it relates to business involvement in development ○ release management • documentation; • training; • support; • service levels. |
| | | <p>Describe how software development is conducted within governance structures and the role of the project manager.</p> |
| | | <p>Understand how effective team-working contributes to the effective delivery of software projects.</p> <ul style="list-style-type: none"> • decision making; • conflict resolution; • collaboration; • communication; • peer review and retrospectives. |
| | <p>Understand how to follow a set of functional and non-functional requirements.</p> | <p>Understand how to follow a set of functional and non-functional requirements.</p> |

| | | |
|--|---|---|
| | | <p>Understand the difference between functional and non-functional requirements and how these are used to drive software development activities.</p> <ul style="list-style-type: none"> • how to review requirements; • how to assess their validity; • how they are used as input to software design; • how they are used during testing to ensure adequate test coverage. |
| | | <p>Identify the different types of non-functional requirements, and the reasons they are important to the end-product of software development.</p> <ul style="list-style-type: none"> • availability; • capacity; • performance; • scalability; • reliability; • maintainability; |
| | | <p>Recognise common ways in which software requirements can be expressed.</p> <ul style="list-style-type: none"> • requirements documents – clear, unambiguous; • user stories; • use case diagrams; • process models / flow diagrams; • UML diagrams. |
| | | <p>Describe the qualities of good requirements and the impact of poor requirements.</p> |
| | | <p>Explain how to determine the correct level of test coverage based on each requirement / type of requirement.</p> |
| | <p>Understand the end-user context for the software development activity.</p> | <p>Understand and recognise the relationship between the user and the environment in which the software will be used.</p> |

| | | |
|--|--|--|
| | | <p>Understand the individual business and external constraints and dependencies that need to be taken into account when developing software.</p> <ul style="list-style-type: none"> • compliance; • ethics; • governance; • legality |
| | | <p>Describe the methods used to identify end-user needs.</p> <ul style="list-style-type: none"> • questionnaires; • user interviews; • contextual enquiry; • focus groups; • personas; • customer journey mapping. |
| | <p>Appreciate the importance of seamlessly connecting applications to databases and understand types of data storage and their applications.</p> | <p>Explain the purpose of data storage for storing new information (orders or customer information).</p> <ul style="list-style-type: none"> • orders; • customer information. |
| | | <p>Explain the purpose of data storage for extracting and displaying data.</p> <ul style="list-style-type: none"> • products; • pricing. |
| | | <p>Explain the concept and key features of databases and data stores.</p> <ul style="list-style-type: none"> • relational databases; • SQL and NoSQL; • data files; • data structures (tables, records, fields, definitions); • document; • key-value. |
| | <p>Demonstrate knowledge of database normalisation.</p> | <p>Explain the purpose and importance of effective data modelling and normalisation.</p> |

| | | |
|---|--|--|
| | | Demonstrate the principle of normalisation, that information or data should be stored only once. |
| Understand why there is a need to follow good coding practices. | | <p>Explain the importance of good coding practice.</p> <ul style="list-style-type: none"> • quality of coding development. <ul style="list-style-type: none"> ○ design documentation ○ structure of code ○ consistent design and structure ○ secure code |
| | | <p>Explain the purpose of good software coding principles and practices.</p> <ul style="list-style-type: none"> • the basic common principles; <ul style="list-style-type: none"> ○ DRY (don't repeat yourself) • defensive programming; • commenting; • refactoring; • patterns / anti-patterns. |
| | | Understand that there are a range of open and organisational coding standards and where to source them. |
| Understand the principles of good interface design. | | <p>Explain human computer interaction and understand the issues associated with interactive systems.</p> <ul style="list-style-type: none"> • usability / ease of use and intuitive design; • graphical user interfaces (GUI) for different types of devices; • ergonomic design. |

| | | |
|--|--|---|
| | | <p>Describe the key concepts and processes of good user interface design.</p> <ul style="list-style-type: none"> • design principles; • design patterns; • tools; <ul style="list-style-type: none"> ○ wireframes ○ prototypes • techniques and methods. <ul style="list-style-type: none"> ○ A/B testing |
| | | <p>Explain the importance of usability when developing interactive systems.</p> |
| | | <p>Describe the fundamental considerations for developing an accessible system and the purpose of the Web Accessibility Initiative (WAI).</p> |
| | <p>Understand the importance of building security in to software at the development stage.</p> | <p>Describe the following types of security issues and the scale and nature of threats that can impact software development.</p> <ul style="list-style-type: none"> • common security attacks; • security versus resilience; • social engineering. <p>Explain what is meant by 'building security in', in terms of secure software development and creating a secure end-product, and why it is important.</p> <ul style="list-style-type: none"> • the role coders play in determining a secure software end-product; • the impact they can have on security by not building security in; • why building security in at the start is better than trying to retrofit later. |

| | | |
|--|--|---|
| | | <p>Describe proactive security approaches during software design and development.</p> <ul style="list-style-type: none">• security development lifecycle (SDLC);• defensive design / defensive programming;• test creation and execution;• permission setting and role based access;• physical infrastructure and security. |
| | | <p>Explain approaches to make software more secure.</p> <ul style="list-style-type: none">• security scanning;• penetration testing;• fuzzing;• load testing. |

5. Assessment

5.1 Summary of assessment methods

The qualification is assessed in controlled exam conditions using a one-hour multiple-choice examination consisting of 40 questions.

The exams are externally marked.

5.2 Availability of assessments

To be able to offer BCS Qualifications you need to become a BCS Approved Training Provider.

All staff members who are involved in the management, invigilation and training must be registered with BCS. Suitably qualified individuals may be registered for more than one role. At least two members of staff must be registered with BCS in one of the roles in order for the Training Provider to retain Training Provider approval.

5.3 Grading

The exam has a pass mark of 65%.

5.4 Externally assessed units

External tests from BCS come in the form of automated tests. The tests offer instant results to the learner.

5.5 Specimen assessment materials

A sample test is available on the BCS Website.

5.6 Support materials

BCS provides the following resources specifically for these qualifications:

| Description | How to access |
|--------------------|----------------------|
| Syllabus | Available on website |
| Sample tests | Available on website |

5.7 Access to Assessment

BCS seeks to provide equal Access to Assessment for all learners, ensuring that there are no unnecessary barriers to assessment and that any reasonable adjustments for learners preserve the validity, reliability and integrity of the qualification.

We will consider requests from BCS approved Training Providers for reasonable adjustments and special considerations to be approved for a learner. The decision will be based on the individual needs of the learner as assessed by suitably qualified professionals. In promoting this policy, BCS aims to ensure that a learner is not disadvantaged in relation to other learners and their certificate accurately reflects their attainment.

6. Contact Points

BCS Qualifications Client Services is committed to providing you with professional service and support at all times through a single, dedicated point of contact. With a flexible and proactive approach, our team will work together with you to ensure we deliver quality solutions that are right for you.

BCS, The Chartered Institute for IT
First Floor, Block D, North Star House, North Star Avenue,
Swindon SN2 1FA

T: +44 (0) 1793 417 424

W: www.bcs.org/qualifications

If you require this document in accessible format, please call +44 (0) 1793 417 424

© BCS, The Chartered Institute for IT, is the business name of The British Computer Society (registered charity no. 292786).