BCS Level 2 Thinking as a Coder Award

Qualification Guide

This qualification is regulated by one or more of the following: Ofqual, Qualifications Wales, CCEA Regulation or SQA.

QAN: 610/0517/4

January 2023 v1.0

CONTENTS

bcs
The Chartered Institute for IT

Introduction

Coding is becoming the new standard of literacy, with skills used in areas as varied as art and design, engineering, data analysis, and science. The new Level 2 Thinking as a Coder qualification from BCS and ICDL explores essential computational thinking and coding skills to support learners in creating simple computer programs. Computational thinking is helpful in many contexts, not just programming and software development; as part of this module, learners will discover that their skills are relevant to diverse areas such as problem-solving, pattern recognition, abstraction, and algorithms.

Computing is a broad field, and its applications continue to grow. Building on this everyday experience of technology, the new Level 2 Thinking as a Coder qualification from BCS and ICDL aims to develop learners' understanding of key concepts in computing and the typical activities involved in creating programs, as well as using problem decomposition, applying project management methodologies such as test, debug, and release, and other vital skills. Alongside a solid foundation in coding, the qualification also emphasises the importance of teamwork, promoting communication and collaboration with other learners, and providing them with transferrable skills to help them stand out in the job market.

Exploring coding inspires a lifelong love of learning, creativity and logic, preparing learners for life in an exciting, ever-changing digital world.

BCS, The Chartered Institute for IT

As the Chartered Institute for IT, we are the digital specialists and the only awarding body focussed on computing and IT. Our commitment under our royal charter is to ensure everyone within society has access to the basic skills required to live and work in a digital age.

Qualification Suitability and Overview

The Level 2 Thinking as a Coder qualification has been designed for learners aged 11 and above.

Examples of learners who may benefit from this qualification include those interested in further qualifications in computing (e.g. GCSE Computer Science) or those aspiring to a future career in the digital or technology sectors.

At the end of this qualification, learners will be able to:

- Understand key concepts relating to computing and the typical activities involved increating a program.
- Understand and use computational thinking techniques like problem decomposition, pattern recognition, abstraction and algorithms to analyse a problem and develop solutions.
- Write, test and modify algorithms for a program using flowcharts and pseudocode.
- Understand key principles and terms associated with coding and the importance of well-structured and documented code.
- Understand and use programming constructs like variables, data types, and logic in a program.
- Improve efficiency and functionality by using iteration, conditional statements, procedures and functions, as well as events and commands in a program.
- Test and debug a program and ensure it meets requirements before release.

In order to complete this Level 2 qualification, learners will have to complete all five mandatory modules. An overview of these can be found below, while further details are included under Module Criteria.

MANDATORY UNITS	LEVEL
Computational Terms Learners will explore key computing concepts.	2
Computational Thinking Learners will explore problem analysis and algorithms.	2
Starting to Code Learners will explore variables and data types.	2
Building Using Code Learners will explore logic, iteration, conditionality, procedures and functions, and events and commands.	2
Test, Debug and Release Learners will explore how to check for and fix errors.	2

LEVEL 2 THINKING AS A CODER AWARD	
QAN	610/0517/4
Entry Requirements	N/A
Guided Learning Hours (GLH)	39 hours
Total Qualification Time (TQT)	70 hours
Assessment Method	45-minute online test with 36 questions.
Outcome	Pass/fail. The pass mark for the test is 75%.

Although there are no formal entry requirements for this qualification, it would be beneficial for learners to have a basic understanding of IT, such as that acquired in IT lessons at school, and an interest in coding. The L2 Thinking as a Coder qualification also links nicely to Computing topics which learners may previously have studied or may go on to study in Key Stage 4 at GCSE or Key Stage 5 as part of a post-16 qualification.

Offering Level 2 Thinking as a Coder

To offer this qualification, an organisation must be a BCS Approved Centre.

Details of what is required to be a centre can be found on our website.

Learner Progression

This Level 2 Thinking as a Coder Award can be seen as a direct progression route for a learner who has completed the Level 1 Robotics qualification. After completing Level 2 Thinking as a Coder, we hope that all of our learners will feel equipped and inspired to go on to study further digital or IT qualifications in the future.

Learners who are particularly interested in continuing to explore the world of coding might like to progress to the BCS Level 3 Award in Principles of Coding, part of the Level 3 Digital Marketer apprenticeship. Other Level 3 apprenticeships from BCS' Digital Apprenticeships may also interest learners following Thinking As A Coder, such as Level 3 Software Development Technician.You can find out more about our Digital Apprenticeships on the BCS website.

Businesses in the technology sector have also spoken about the need to improve basic IT skills. Learners wanting to work on this core area may be interested in our Essential Digital Skills Qualifications (EDSQ). BCS also offers the International Certification of Digital Literacy (ICDL) Level 2 Award and Level 2 Certificate, which employers worldwide recognise as the benchmark in digital and IT user skills.

Module Criteria

COMPUTING TERMS	
Skill set:	**Assessment Criteria : The learner can...**
Key Concepts	Define the term computing.
	Define the term computational thinking.
	Define the term program.
	Define the term code. Distinguish between source code, machine code.
	Understand the terms program description and specification.
	Recognise typical activities in the creation of a program: analysis, design, programming, testing, enhancement.
	Understand the difference between a formal language and a natural language.

COMPUTATIONAL THINKING METHODS	
Skill set:	**Assessment Criteria : The learner can...**
Problem Analysis	Outline the typical methods used in computational thinking: decomposition, pattern recognition, abstraction, algorithms.
	Use problem decomposition to break down data, processes, or a complex problem into smaller parts.
	Identify patterns among small, decomposed problems.
	Use abstraction to filter out unnecessary details when analysing a problem.
	Understand how algorithms are used in computational thinking.
Algorithms	Define the programming construct term sequence. Outline the purpose of sequencing when designing algorithms.
	Recognise possible methods for problem representation like: flowcharts, pseudocode.
	Recognise flowchart symbols like: start/stop, process, decision, input/output, connector, arrow.
	Outline the sequence of operations represented by a flowchart, pseudocode.

COMPUTATIONAL THINKING METHODS (CONTINUED)	
Skill set:	**Assessment Criteria : The learner can...**
Algorithms	Write an accurate algorithm based on a description using a technique like: flowchart, pseudocode.
	Fix errors in an algorithm like: missing program element, incorrect sequence, incorrect decision outcome.

STARTING TO CODE	
Skill set:	**Assessment Criteria : The learner can...**
Getting Started	Describe the characteristics of well-structured and documented code like: indentation, appropriate comments, descriptive naming.
	Use simple arithmetic operators to perform calculations in a program: +, -, /, *.
	Understand the precedence of operators and the order of evaluation in complex expressions. Understand how to use parenthesis to structure complex expressions.
	Understand the term parameter. Outline the purpose of parameters in a program.
	Define the programming construct term comment. Outline the purpose of a comment in a program.
	Use comments in a program.
Variables and Data Types	Define the programming construct term variable. Outline the purpose of a variable in a program.
	Define and initialise a variable.
	Assign a value to a variable.
	Use appropriately named variables in a program for calculations, storing values.
	Use data types in a program: string, character, integer, float, Boolean.
	Use an aggregate data type in a program like: array, list, tuple.
	Use data input from a user in a program.
	Use data output to a screen in a program.

BUILDING USING CODE	
Skill set:	**Assessment Criteria : The learner can...**
Logic	Define the programming construct term logic test. Outline the purpose of a logic test in a program.
	Recognise types of Boolean logic expressions to generate a true or false value like: =, >, <, >=, <=, <>, !=, ==, AND, OR, NOT.
	Use Boolean logic expressions in a program.
Iteration	Define the programming construct term loop. Outline the purpose and benefit of looping in a program.
	Recognise types of loops used for iteration: for, while, repeat.
	Use iteration (looping) in a program like: for, while, repeat.
	Understand the term infinite loop.
	Understand the term recursion.
Conditionality	Define the programming construct term conditional statement. Outline the purpose of conditional statements in a program.
	Use IF...THEN...ELSE conditional statements in a program.
Procedures and Functions	Understand the term procedure. Outline the purpose of a procedure in a program.
	Write and name a procedure in a program.
	Understand the term function. Outline the purpose of a function in a program.
	Write and name a function in a program.
Events and Commands	Understand the term event. Outline the purpose of an event in a program.
	Use event handlers like: mouse click, keyboard input, button click, timer.
	Use available generic libraries like: math, random, time.

TEST, DEBUG AND RELEASE	
Skill set:	**Assessment Criteria : The learner can...**
Run, Test and Debug	Understand the benefits of testing and debugging a program to resolve errors.
	Understand types of errors in a program like: syntax, logic.
	Run a program.
	Identify and fix a syntax error in a program like: incorrect spelling, missing punctuation.
	Identify and fix a logic error in a program like: incorrect Boolean expression, incorrect data type.
Release	Check your program against the requirements of the initial description.
	Describe the completed program, communicating purpose and value.
	Identify enhancements, improvements to the program that may meet additional, related needs.

Resources

Many valuable resources are available to help you and your learners make the most of the Level 2 Thinking As a Coder qualification. These are available from Skillsbox and the Atlas Cloud platform.

AVAILABLE RESOURCES

Diagnostic test

This test enables learners to practise and determine if they are ready to progress to the assessment. It is accessed via the Skillsbox online platform. There are 55 questions in different formats, including:

- Multiple-choice.
- Drag-and-drop.
- Hotspot.

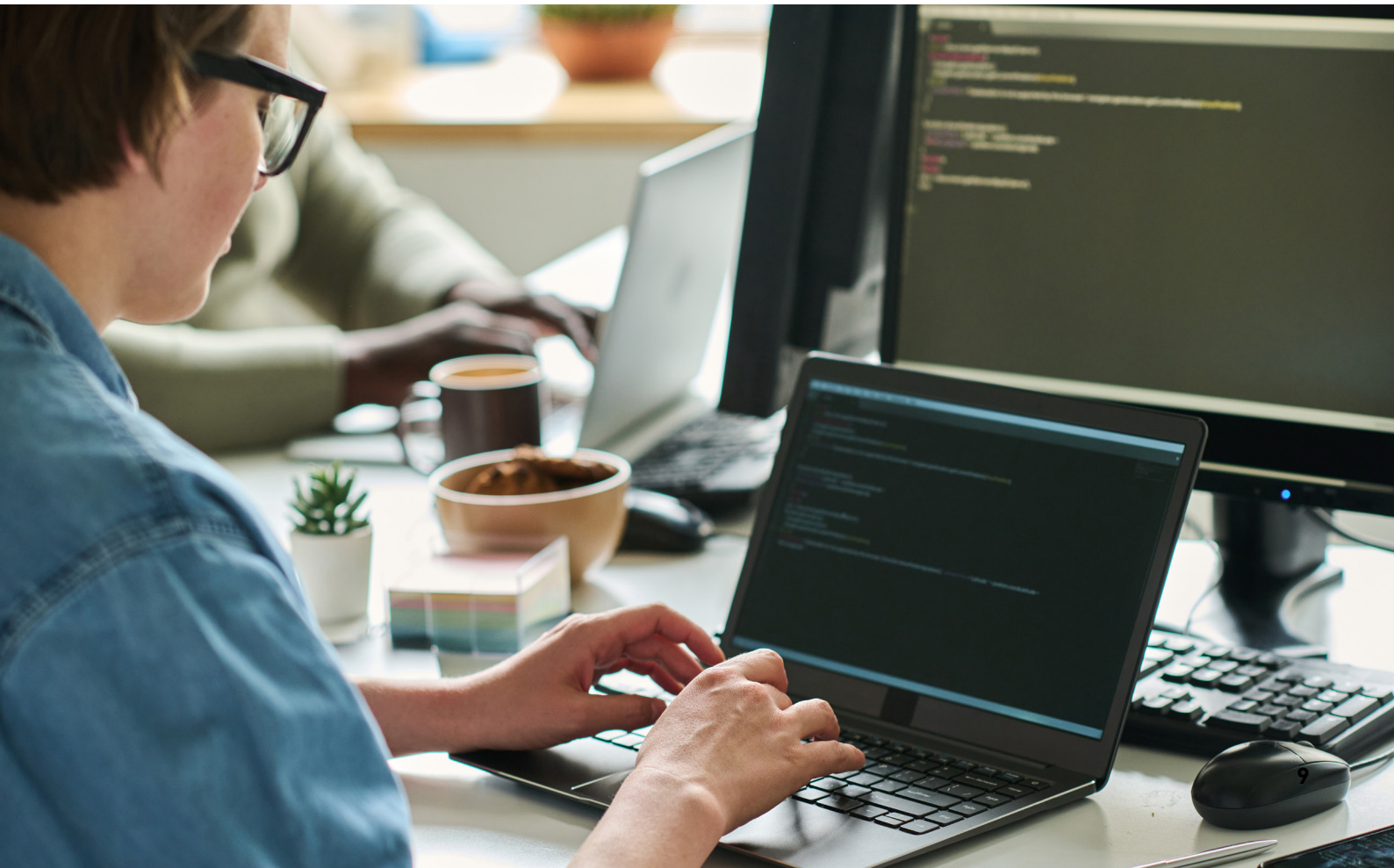
Learner material eBook

The eBook contains 15 lessons, each with a set of criteria that the learner should meet once the lesson is complete. At the end of each lesson, there is a multiple-choice review exercise ranging from 2 to 7 questions.

The eBook also includes a mapping table to show which areas of the syllabus correspond to which lesson.

Teacher handbook

The handbook includes an overview of computing, a background in Computational Thinking, a summary of Python, and how to install Python. The handbook also contains learning objectives, lesson plans, additional resources, exercises, answers to review questions in the learner material eBook, and a guided lesson planner.



Assessment

Online test

The online test assesses the competencies outlined in the ICDL Thinking as a Coder syllabus. Learners must be registered to the Thinking as a Coder module to take a test.

The online test is invigilated and takes place in a registered test centre, which in many cases may be the school. The test is delivered through Skillsbox, an automated test system.

A variety of question types are used in the test to test the learner's mastery of the knowledge and skills outlined in the curriculum. The question type and format are primarily determined by the type of knowledge or skill being measured. Question types may include multiple choice, drag and drop, hotspots, match-ups, fill-in-the-blanks, or practical in-application tasks.

The test takes 45 minutes, and the passing mark is 75%. Learners must pass to complete the qualification.

While BCS would generally want to avoid making changes to either grade thresholds or grading algorithms, there is potential for them to change to maintain standards.

Reasonable Adjustments

Centres will receive guidance on reasonable adjustments under Equalities Law, including, but not exclusively, ensuring an environment which will allow access by a disabled learner or to make alternative arrangements such as a different venue or different equipment suitable for the learner.

Outcomes and Reassessment

When a learner completes the online test using the Skillsbox platform, the results are submitted directly to BCS.

Resits are available for this qualification.

Appeals

If situations arise that call into the question the validity of an awarding decision, for example, via an appeal or an enquiry in accordance with our Appeals Policy, or an error has been made and a learner has incorrectly been awarded, or not awarded, a qualification achievement issue will be brought to the attention of the Service Delivery Manager - Qualifications. Our Appeals Policy is available from the Approved Centre Forum.

Skillsbox

Accessing the online assessments

The test may be completed via the Skillsbox online platform on an on-demand basis. Centres will have access to add and manage users and tests.

You can access Skillsbox by logging in **here**.

System Requirements

SYSTEM CHECK	REQUIREMENTS	ADDITIONAL INFORMATION
Operating System	Windows 7/8/10	Only Microsoft Windows is supported for in-application testing
Browser	Internet Explorer 11 Firefox Google Chrome	A plugin is required for in-application testing
Plugin Installation	PSI in-application Plugin is required for tests	All Supported Browsers: Ensure the plugin is fully installed and detected. Additional Chrome Requirements: Ensure the extension has been installed Additional Firefox Requirements: Ensure the Firefox extension and the plugin are installed
.NET Framework	.NET 3.X Framework is required	.NET 3.X framework is required for applications to run**
Microsoft Office	Microsoft Office applications must be installed.	In-application testing will not work with browser versions of Office365
Access to Work Files (Z:/)	Skillsbox Atlas Cloud uses a drive mapping script to create Z:/ on the machine to store test files.	The mapped drive must be visible to candidates if there is already a Z:/ drive on the network the script will work backwards to find the next available letter to map the drive to.
Registry Access	User must have read/write access to HKEY_CURRENT_USER	This is default in Windows

Further guidance around using Skillsbox can be found here on the BCS website.

Frequently Asked Questions

Q) How long does this qualification take to complete?

A) This qualification has 39 guided learning hours, and a total qualification time of 70 hours.

Q) What learning materials or courseware are available?

A) Learners will be able to access a diagnostic test via the Skillsbox online platform, to practise and determine if they are ready to progress to the second part of the assessment, the test. There are also two handbooks available: one for learners and one for teachers. The learner material handbook will be accessible as an e-book on Atlas Cloud platform. The teacher handbook includes an overview of computational thinking, planning considerations, lesson plans including learning outcomes, resources needed, activity description, and useful links.

Q) Can this qualification be delivered remotely?

A) All the tests are online, so it is possible to deliver remotely. BCS recommend centres use their discretion when considering to deliver remotely and consider the age range of their learners when making any decision.

Q) What is GLH and TQT?

A) Guided Learning Hours (GLH) indicates the approximate time (in hours) that the learner will be supervised during any teaching, learning or assessment activities.

Total Qualification Time (TQT) is a predication of the total time a learner with no prior knowledge might need to complete the course. TQT is made up of two elements: GLH, and all other hours (an estimate of the number of hours a learner will reasonably spend on any unsupervised learning or assessment activities including homework, research, exam preparation and formal assessment) so that they can successfully achieve the qualification.

Q) What practice tests are available?

A) A diagnostic test is available through the Skillsbox platform.

Glossary

We recognise that Level 2 Thinking as a Coder may not just be new for learners – it may be a new topic area for teachers, schools and centres too. With this in mind, we have included a glossary of some key terms used both within this Qualification Guide and in ICDL's Thinking as a Coder Teacher Handbook.

Abstraction	A process of filtering out characteristics of patterns which are not needed.
Aggregate data type	Any types of data that can be referenced as a single entity, but that also consist of more than one piece of data, e.g., strings, arrays, classes, structures.
Algorithm	A plan; a set of step-by-step instructions to solve a problem.
Arithmetic operator	A mathematical function that is used to perform a calculation, e.g. add, subtract, multiple, divide.
Boilerplate code	Sections of code that are repeated in multiple places with little to no variation.
Boolean	A data type where the data only has two possible variables: true or false, e.g. Boolean logic is a form of algebra where all values are either true or false.
Computational thinking	A problem-solving process which allows us to take a complex problem, understand what the problem is and develop possible solutions. Its four cornerstones are decomposition, pattern recognition, abstraction, and algorithms.
Conditional statement	Evaluates an expression and executes instructions depending on the evaluation's outcome.
Debug	Finding mistakes (bugs) in code and getting rid of them.
Event handlers	Programs that respond to events.
Formal language	Consists of words whose letters are taken from an alphabet and are well-formed according to a specific set of rules. A formal language's alphabet consists of symbols, letters or tokens.
Function	A section of code that can be called by another part of the program, so that it returns one single value.
Iteration	Repeating steps, or instructions, over and over; often called a loop.
Machine code	Consists of binary or hexadecimal numbers.
Parameter	Allow us to pass information or instructions into functions and procedures.
Pattern recognition	Involves finding the similarities or patterns among small, decomposed problems that can help us solve more complex problems more efficiently.
Problem decomposition	Looking for patterns among and within the smaller problems that make up the complex problem.

Procedure	A small section of a program that performs a specific task.
Pseudocode	A method of writing up a set of instructions for a computer programme using plain English; a good way of planning a program before coding.
Python	A high-level, general-purpose programming language.
Recursion	A programming technique using function or algorithm that calls itself one or more times until a specified condition is met.
Source code	Occurs when the code given does not follow the syntax rules of the programming language, e.g. misspelling a statement or missing brackets.
Variables	Data values that can change when the user is asked a question.

CONTACT

For any queries relating to this document or the delivery of this qualifcation, contact;

T: 01793 417445 **E**: bcssales@bcs.uk

If you have any technical issues running the online assessments, please contact;

Skillsbox Support – support@skillsbox.com

For further information please contact:

BCS
The Chartered Institute for IT
3 Newbridge Square
Swindon
SN1 1BY

T +44 (0)1793 417 445

www.bcs.org