



# **Teddington Sixth Form**

# **A Level Computer Science**

## **Course Details & Transition Tasks**

## **2020-2022**

[A level Computer Science](#)  
[Exam Board – AQA](#)

[Year 1 \(Year 12\)](#)

This booklet is designed to give you all the information you need before starting the Computer Science course.

Course Title	Size & Structure
AQA Computer Science	2 Units Paper 1 : On screen exam (50% of AS) Paper 2 : Written exam (50% of AS)

**Paper 1**

This paper tests a student's ability to program, as well as their theoretical knowledge of computer science from the subject content below.

On-screen exam : 1 hour 30 minutes                      50% of AS

Students answer a series of short questions and write/adapt/extend programs in an electronic answer document provided by the exam board. AQA will issue preliminary material, a skeleton program and, where appropriate, test data, for use in the exam.

*[Section 1: Fundamentals of Programming](#)*

- Data types
- Programming structures
- Arithmetic, Relational & Boolean operations
- Constants and variables
- String-handling
- Random numbers
- Exception handling
- Subroutines

*[Section 2: Fundamentals of Data Structures](#)*

- Data structures
- Arrays
- Fields, records and files

*[Section 3: Systematic Approach to Problem Solving](#)*

- Analysis
- Design
- Implementation
- Testing
- Evaluation

*[Section 4: Theory of computation](#)*

- Problem-Solving
- Following and writing algorithms
- Abstraction & Decomposition
- Composition
- Automation

- Finite State Machines

## **Paper 2**

This paper tests a student's ability to answer questions from subject content below.

Students answer a series of short-answer and extended-answer questions.

### *Section 5 : Fundamentals of Data Representation*

- Numbers – natural, integer, rational, irrational, real, ordinal, counting and measuring
- Number Bases
- Units of information – bits, bytes & units
- Binary Number System – unsigned, signed arithmetic, two's complement, fractions
- Information coding systems – character forms, ASCII, Unicode, Error checking
- Representing images, sound and other data

### *Section 6: Fundamentals of Computer Systems*

- Hardware and software
- Classification of software
- System software
- Role of an operating system
- Classification of programming languages
- Types of program translator
- Logic Gates
- Boolean Algebra

### *Section 7 : Fundamentals of Computer Organisation and Architecture*

- Internal hardware components
- Stored program concept
- Structure and role of processor and its components
- Fetch-Execute cycle and the role of registers
- Processor instruction set and addressing modes
- Machine code / Assembly language
- External hardware devices

### *Section 8 : Consequences of uses of Computing*

- Moral, ethical, legal and cultural issues and opportunities

### *Section 9 : Fundamentals of Communication and Networking*

- Communication methods and basics
- Networking topologies
- Wireless networking

## Year 2 (Year 13)

Course Title	Size & Structure
AQA A Level Computer Science	3 Units Paper 1 : On screen exam (40%) Paper 2 : Written exam (40%) NEA : Practical project (20%)

### **Paper 1**

This paper tests a student's ability to program, as well as their theoretical knowledge of computer science from the subject content below.

On-screen exam : 2 hour 30 minutes                      40% of A-Level

Students answer a series of short questions and write/adapt/extend programs in an electronic answer document provided by the exam board. AQA will issue preliminary material, a skeleton program and, where appropriate, test data, for use in the exam.

### *Section 10: Fundamentals of Programming*

- Data types
- Programming structures
- Arithmetic, Relational & Boolean operations
- Constants and variables
- String-handling
- Random numbers
- Exception handling
- Subroutines
- Recursion
- Programming paradigms

### *Section 11: Fundamentals of Data Structures*

- Data structures
- Arrays
- Fields, records and files
- Abstract data types/structures
- Queues, Stacks, Graphs, Trees, Hash tables, Dictionaries, Vectors

### *Section 12: Fundamentals of Algorithms*

- Graph-traversal
- Tree-traversal
- Reverse polish
- Searching, sorting & optimisation algorithms

### *Section 13 : Theory of Computation*

- Abstraction and automation
- Regular languages
- Context-free languages
- Classification of algorithms
- Models of computation

## Paper 2

This paper tests a student's ability to answer questions from subject content below.

Written exam : 2 hour 30 minutes                      40% of A-Level

Students answer a series of short-answer and extended-answer questions.

### *Section 14: Fundamentals of Data Representation*

- Numbers – natural, integer, rational, irrational, real, ordinal, counting and measuring
- Number Bases & Units of information – bits, bytes & units
- Binary Number System – unsigned, unsigned arithmetic, two's complement, fractions
- Floating point normalisation, underflow and overflow
- Information coding systems – character forms, ASCII, Unicode, Error checking
- Representing images, sound and other data

### *Section 15: Fundamentals of Computer Systems*

- Hardware and software
- System software & Role of an operating system
- Classification of programming languages
- Types of program translator
- Logic Gates & Boolean Algebra

### *Section 16: Fundamentals of Computer Organisation and Architecture*

- Internal hardware components
- Stored program concept
- Structure and role of processor and its components
- Fetch-Execute cycle and the role of registers
- Processor instruction set and addressing modes
- Machine code / Assembly language
- External hardware devices

### *Section 17: Consequences of uses of Computing*

- Moral, ethical, legal and cultural issues and opportunities

### *Section 18: Fundamentals of Communication and Networking.*

- Communication methods and basics
- Networking topologies & Wireless networking
- The Internet and TCP/IP

### *Section 19: Fundamentals of Databases*

- Conceptual data models and entity relationship modelling
- Relational databases
- Structured Query Language (SQL)

### *Section 20: Big Data*

- Big Data

### *Section 21: Fundamentals of Functional Programming*

- Functional programming paradigm
- Writing functional programs
- Lists in functional programming

## Non-Exam Assessment (NEA)

The non-exam assessment assesses student's ability to use the knowledge and skills gained through the course to solve or investigate a practical problem. Students will be expected to follow a systematic approach to problem solving

Practical project: 75 marks                      20% of A-Level

### *Section 22: The Computing Practical Project*

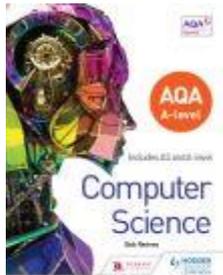
- Analysis
- Design
- Implementation
- Testing
- Evaluation

The project allows students to develop their practical skills in the context of solving a realistic problem or carrying out an investigation. The project is intended to be as much a learning experience as a method of assessment; students have the opportunity to work independently on a problem of interest over an extended period, during which they can extend their programming skills and deepen their understanding of computer science.

The most important skill that should be assessed through the project is a student's ability to create a programmed solution to a problem or investigation. This is recognised by allocating 42 of the 75 available marks to the technical solution and a lower proportion of marks for supporting documentation to reflect the expectation that reporting of the problem, its analysis, the design of a solution or plan of an investigation and testing and evaluation will be concise.

## Textbooks and Resources

Textbook (students not required to purchase)



AQA A-level Computing Student Book

Author: Bob Reeves

Publisher: Hodder Education

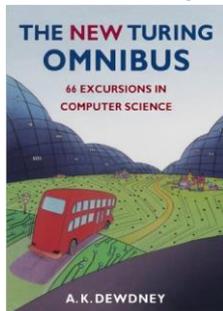
ISBN-13: 978-1-4718-3951-1

Software (required)

Python 3.x - <https://www.python.org/downloads/>

Pyzo IDE - <http://www.pyzo.org/index.html>

## Wider Reading:

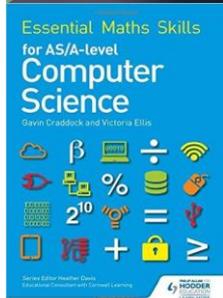


New Turing Omnibus

Author: A. k. Dewdney

Publisher: Palgrave Macmillan

ISBN-13: 978-0805071665

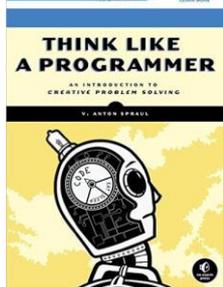


Essential Maths Skills for AS/A Level Computer Science

Author: Gavin Craddock & Victoria Ellis

Publisher: Philip Allan

ISBN-13: 978-1471863578

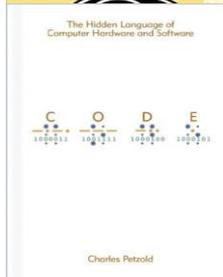


Think Like a Programmer: An Introduction to Creative Problem Solving

Author: V. Anton Sprual

Publisher: No Starch Press

ISBN-13: 978-1593274245



Code: The Hidden Language of Computer Hardware and Software

Author: Charles Petzold

Publisher: Microsoft Press

ISBN-13: 978-0735611313

## Transition Tasks

One of the primary obstacles to success in Computer Science at A Level is difficulty with programming. There is a far greater amount of programming required than at GCSE, and the complexity of the tasks you are asked to do is much higher. The nature of the exam being computer-based means that if you cannot program confidently, then you will struggle to get a grade that you are happy with. Many of the tasks that you would attempt during year 11 of your GCSE, are simply the starting point for a more complex exercise in year 12 or 13.

To this end, the most productive way for you to prepare for A Level Computer Science is to develop your coding skills so that you are confident, competent and able to take on the challenge of A Level programming.

<https://automatetheboringstuff.com/>

<https://inventwithpython.com/cracking/>

Week	Book	Chapters
1	Automate the Boring Stuff	Chapter 0 – <a href="#">Introduction</a> Chapter 1 – <a href="#">Python Basics</a> Chapter 2 – <a href="#">Flow Control</a> Chapter 3 – <a href="#">Functions</a> Chapter 4 – <a href="#">Lists</a>
2	Automate the Boring Stuff	Chapter 5 – <a href="#">Dictionaries and Structuring Data</a> Chapter 6 – <a href="#">Manipulating Strings</a> Chapter 7 – <a href="#">Pattern Matching with Regular Expressions</a> Chapter 8 – <a href="#">Input Validation</a>
3	Automate the Boring Stuff	Chapter 9 – <a href="#">Reading and Writing Files</a> Chapter 10 – <a href="#">Organizing Files</a> Chapter 11 – <a href="#">Debugging</a> Chapter 12 – <a href="#">Web Scraping</a>
4	Automate the Boring Stuff	Chapter 13 – <a href="#">Working with Excel Spreadsheets</a> Chapter 14 – <a href="#">Working with Google Spreadsheets</a> Chapter 15 – <a href="#">Working with PDF and Word Documents</a> Chapter 16 – <a href="#">Working with CSV Files and JSON Data</a>
5	Cracking Codes with Python	<a href="#">Introduction</a> Chapter 1 - <a href="#">Making Paper Cryptography Tools</a> Chapter 2 - <a href="#">Programming in the Interactive Shell</a> Chapter 3 - <a href="#">Strings and Writing Programs</a> Chapter 4 - <a href="#">The Reverse Cipher</a>
6	Cracking Codes with Python	Chapter 5 - <a href="#">The Caesar Cipher</a> Chapter 6 - <a href="#">Hacking the Caesar Cipher with Brute-Force</a> Chapter 7 - <a href="#">Encrypting with the Transposition Cipher</a> Chapter 8 - <a href="#">Decrypting with the Transposition Cipher</a> Chapter 9 - <a href="#">Programming a Program to Test Your Program</a>
7	Cracking Codes with Python	Chapter 16 - <a href="#">Programming the Simple Substitution Cipher</a> Chapter 17 - <a href="#">Hacking the Simple Substitution Cipher</a> Chapter 18 - <a href="#">Programming the Vigenere Cipher</a> Chapter 19 - <a href="#">Frequency Analysis</a> Chapter 20 - <a href="#">Hacking the Vigenere Cipher</a>