Contents

Teacher guidance
Notes on course delivery  v
Two-year course schedule  vi
Two-stream course schedule  viii
Curriculum coverage  xi
Using Python xvii
Introduction to Kerboodle xviii

Lesson plans

1 Data representation
1.1 Binary data 1
1.1 Bits and bytes 2
1.1 Binary and denary 3
1.1 Counting in binary 4
1.1 Convert binary to denary 5
1.1 Convert denary to binary 6
1.2 What is hexadecimal? 7
1.2 Hexadecimal and denary 8
1.2 Hexadecimal and binary 9
1.2 How hexadecimal is used 10
1.3 Digital data 12
1.3 Digital graphics 13
1.3 Digital sound and video 14
1.3 Compression 15

2 Communications and the Internet
2.1 How data is transmitted 17
2.1 Serial or parallel? 18
2.1 Data bus 19
2.1 Transmission errors 20
2.1 Parity check 21
2.1 Check digit and checksum 22
2.2 What is the Internet? 24
2.2 What is the World Wide Web? 25
2.2 HTML 26
2.2 HTTP: Hypertext transfer protocol 27
2.2 TCP/IP 28
2.3 Staying safe 29
2.3 Malware and hacking 30
2.3 Protective software 31

3 The processor
3.1 Electronic processing 32
3.1 The NOT gate 33
3.1 The AND gate 34
3.1 The OR and XOR gates 35
3.1 The NAND and NOR gates 36
3.2 Logic statements 37
3.2 Simplify statements 38
3.2 Logic circuits 39
3.2 Truth tables and circuits 40
3.2 Truth tables (continued) 41
3.2 Solve a problem 42
3.2 Repeat inputs 43
3.3 The central processing unit (CPU) 44
3.3 The fetch-execute cycle 45
3.3 Registers and buses 46

4 Hardware
4.1 Keyboard and mouse 47
4.1 Touch screens 48
4.1 Camera and microphone 49
4.1 Barcode readers 50
4.1 Scanners 51
4.1 Sensors 52
4.1 Control systems 53
4.2 Monitors and display 54
4.2 Printers 55
4.2 Sound 56
4.2 Actuators 57
4.2 Manufacturing objects 58
4.2 Output in real life 59
4.3 Primary storage 60
4.3 Measuring storage 61
4.3 File sizes 62
4.3 Magnetic storage 63
4.3 Optical storage 64
4.3 Solid state (flash) storage 65
4.3 Use and choice of storage 66
# Software

5.1 What is software? 
5.1 Operating systems 
5.1 Functions of an operating system 
5.2 Low-level languages 
5.2 High-level languages 

# Security

6.1 Data security 
6.1 Security threats 
6.1 Malpractice and crime 
6.2 Proof of identity 
6.2 Firewalls 
6.2 Security protocols 
6.2 Encryption 
6.2 Security examples 

# Ethics

7.1 Copyright 
7.1 Free software 
7.1 Hackers and crackers 

# Programming

8.1 Introduction to Python 
8.1 Algorithms 
8.2 Output 
8.2 Sequence 
8.2 Input 
8.2 Assign a value 
8.2 Calculated values 
8.2 Variables in pseudocode 
8.2 Variables in flowcharts 
8.3 Logical decision 
8.3 Python if...

# Solution development

9.1 Count how many 
9.1 Calculate a total 
9.1 Calculate an average 
9.1 Verification 
9.1 Validation 
9.2 Test data 
9.2 Evaluation 
9.3 Trace tables 
9.3 Trace tables (loops) 
9.3 Trace tables (flowcharts) 
9.3 Analyse algorithms 
9.3 Find errors in algorithms 
9.3 Create an algorithm 
9.4 Top-down programming 
9.4 Structure diagrams 
9.4 Code libraries 

# Databases

10.1 Records and fields 
10.1 Data types 
10.1 Primary key 
10.2 Select fields 
10.2 Select records
Curriculum coverage
All elements of the curriculum are delivered by the Student Book. This table shows the book section and individual lesson(s) where each curriculum topic is covered.

<table>
<thead>
<tr>
<th>1.1 Data representation</th>
<th>Section/spread</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1.1 Binary systems</td>
<td>Chapter 1, topic 1</td>
</tr>
<tr>
<td>• recognise the use of binary numbers in computer systems</td>
<td>1.1 Binary data</td>
</tr>
<tr>
<td></td>
<td>1.1 Bits and bytes</td>
</tr>
<tr>
<td></td>
<td>1.1 Binary and denary</td>
</tr>
<tr>
<td></td>
<td>1.1 Counting in binary</td>
</tr>
<tr>
<td>• convert positive denary integers into binary and positive binary integers into denary</td>
<td>1.1 Binary and denary</td>
</tr>
<tr>
<td></td>
<td>1.1 Convert binary to denary</td>
</tr>
<tr>
<td></td>
<td>1.1 Convert denary to binary</td>
</tr>
<tr>
<td>• show understanding of the concept of a byte and how the byte is used to measure memory size</td>
<td>1.1 Bits and bytes</td>
</tr>
<tr>
<td>• use binary in computer registers for a given application (such as in robotics, digital instruments and counting systems)</td>
<td>1.1 Bits and bytes</td>
</tr>
<tr>
<td></td>
<td>3.3 Registers and buses</td>
</tr>
<tr>
<td>1.1.2 Hexadecimal</td>
<td>Chapter 1, topic 2</td>
</tr>
<tr>
<td>• represent positive numbers in hexadecimal notation</td>
<td>1.2 Hexadecimal and denary</td>
</tr>
<tr>
<td>• show understanding of the reasons for choosing hexadecimal notation to represent numbers</td>
<td>1.2 What is hexadecimal?</td>
</tr>
<tr>
<td>• convert positive hexadecimal integers to and from denary (up to four hexadecimal digits)</td>
<td>1.2 Hexadecimal and denary</td>
</tr>
<tr>
<td>• convert positive hexadecimal integers to and from binary (up to 16 bit)</td>
<td>1.2 Hexadecimal and binary</td>
</tr>
<tr>
<td>• represent numbers stored in registers and main memory as hexadecimal</td>
<td>1.2 How hexadecimal is used</td>
</tr>
<tr>
<td></td>
<td>3.3 Registers and buses</td>
</tr>
<tr>
<td>• identify current uses of hexadecimal numbers in computing, such as defining colours in HTML, MAC address, low-level languages</td>
<td>1.2 How hexadecimal is used</td>
</tr>
<tr>
<td></td>
<td>2.2 HTTP Hypertext transfer protocol</td>
</tr>
<tr>
<td></td>
<td>2.2 TCP/IP</td>
</tr>
<tr>
<td></td>
<td>5.1 Functions of an operating system</td>
</tr>
<tr>
<td>1.1.3 Data storage</td>
<td>Chapter 1, topic 3</td>
</tr>
<tr>
<td>• show understanding that sound (music), pictures, video, text and numbers are stored in different formats</td>
<td>1.3 Digital data</td>
</tr>
<tr>
<td></td>
<td>1.3 Digital graphics</td>
</tr>
<tr>
<td></td>
<td>1.3 Digital sound and video</td>
</tr>
<tr>
<td>• identify and describe methods of error detection and correction, such as parity checks, check digits, checksums and automatic repeat requests (ARQ)</td>
<td>2.1 Transmission errors</td>
</tr>
<tr>
<td></td>
<td>2.1 Parity checks</td>
</tr>
<tr>
<td></td>
<td>2.1 Check digit and checksum</td>
</tr>
<tr>
<td>• show understanding of the concept of MIDI files, JPEG files, MP3 and MP4 files</td>
<td>1.3 Digital sound and video</td>
</tr>
<tr>
<td></td>
<td>1.3 Compression</td>
</tr>
<tr>
<td>• show understanding of the principles of data compression (lossless and lossy) applied to music, video, photographs and text files</td>
<td>1.3 Compression</td>
</tr>
</tbody>
</table>
### 1.2 Communication and Internet technologies

#### 1.2.1 Data transmission
- [Chapter 2, topic 1](#) 2.1 How data is transmitted
  - show understanding of what is meant by transmission of data
  - distinguish between serial and parallel data transmission
  - distinguish between simplex, duplex and half-duplex data transmission
  - show understanding of the reasons for choosing serial or parallel data transmission
  - show understanding of the need to check for errors
  - explain how parity bits are used for error detection
  - show understanding of the use of serial and parallel data transmission in USB and IC

#### 1.2.2 Security aspects
- [Chapter 2, topic 3](#) 2.3 Staying safe
  - show understanding of the security aspects of using the Internet and understand what methods are available to help minimise the risks
  - show understanding of the Internet risks associated with malware, including viruses, spyware and hacking
  - explain how anti-virus and other protection software helps to protect the user from security risks

#### 1.2.3 Internet principles of operation
- [Chapter 2, topic 2](#) 2.2 What is the World Wide Web?
  - show understanding of the role of the browser
  - show understanding of the role of ISP
  - show understanding of what is meant by hypertext transfer protocol (http and https) and HTML
  - distinguish between HTML structure and presentation
  - show understanding of the concepts of MAC address, IP address, URL and cookies

### 1.3 Hardware and software

#### 1.3.1 Logic gates
- [Chapter 3 topics 1–2](#) 3.1 Electronic processing
  - use logic gates to create electronic circuits
  - understand and define the functions of NOT, AND, OR, NAND, NOR and XOR (EOR) gates, including the binary output produced from all the possible binary inputs
  - draw truth tables and recognise a logic gate from its truth table
  - recognise and use the standard symbols used to represent logic gates
  - produce truth tables for given logic circuits
  - produce a logic circuit to solve a given problem or written logic statement
<table>
<thead>
<tr>
<th>1.3.2 Computer architecture and the fetch-execute cycle</th>
<th>Chapter 3, topic 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>• show understanding of the basic von Neumann model for a computer system and the stored program concept</td>
<td>3.3 The central processing unit (CPU)</td>
</tr>
<tr>
<td>• describe the stages of the fetch-execute cycle, including the use of registers and buses</td>
<td>3.3 The fetch-execute cycle</td>
</tr>
<tr>
<td>3.3 Registers and buses</td>
<td>3.3 Registers and buses</td>
</tr>
<tr>
<td>1.3.3 Input devices</td>
<td>Chapter 4, topic 1</td>
</tr>
<tr>
<td>• describe the principles of operation (how each device works) of these input devices:</td>
<td></td>
</tr>
<tr>
<td>• 2D and 3D scanners</td>
<td>4.1 Scanners</td>
</tr>
<tr>
<td>• barcode readers, quick response (QR) code readers</td>
<td>4.1 Barcode readers</td>
</tr>
<tr>
<td>• digital cameras, microphones</td>
<td>4.1 Camera and microphone</td>
</tr>
<tr>
<td>• keyboard and mouse</td>
<td>4.1 Keyboard and mouse</td>
</tr>
<tr>
<td>• touch screens, interactive whiteboard</td>
<td>4.1 Touch screens</td>
</tr>
<tr>
<td>• describe how these principles are applied to real-life scenarios, for example: scanning of passports at airports, barcode readers at supermarket checkouts, and touch screens on mobile devices</td>
<td>4.1 Sensors</td>
</tr>
<tr>
<td>• describe how a range of sensors can be used to input data into a computer system, including light, temperature, magnetic field, gas, pressure, moisture, humidity, pH and motion</td>
<td>4.1 Control systems</td>
</tr>
<tr>
<td>• describe how these sensors are used in real-life scenarios, for example: street lights, security devices, pollution control, games, and household and industrial applications</td>
<td></td>
</tr>
<tr>
<td>1.3.4 Output devices</td>
<td>Chapter 4, topic 2</td>
</tr>
<tr>
<td>• describe the principles of operation of the following output devices:</td>
<td></td>
</tr>
<tr>
<td>• inkjet, laser and 3D printers</td>
<td>4.2 Printers</td>
</tr>
<tr>
<td>• 2D and 3D cutters</td>
<td>4.2 Manufacturing objects</td>
</tr>
<tr>
<td>• speakers and headphones</td>
<td>4.2 Sound</td>
</tr>
<tr>
<td>• actuators</td>
<td>4.2 Actuators</td>
</tr>
<tr>
<td>• flat-panel display screens, such as liquid crystal display (LCD) and light-emitting diodes (LED) display; LCD projectors and digital light projectors (DLP)</td>
<td>4.2 Monitors and display</td>
</tr>
<tr>
<td>• describe how these principles are applied to real-life scenarios, for example: printing single items on demand or in large volumes; use of small screens on mobile devices</td>
<td>4.2 Output in real life</td>
</tr>
<tr>
<td>1.3.5 Memory, storage devices and media</td>
<td>Chapter 4, topic 3</td>
</tr>
<tr>
<td>• show understanding of the difference between: primary, secondary and off-line storage and provide examples of each</td>
<td>4.3 Primary storage</td>
</tr>
<tr>
<td>• primary: read only memory (ROM), and random access memory (RAM)</td>
<td>4.3 Primary storage</td>
</tr>
</tbody>
</table>
- secondary: hard disk drive (HDD) and solid state drive (SSD)
- off-line: digital versatile disc (DVD), compact disc (CD), Blu-ray disc, USB flash memory and removable HDD
- describe the principles of operation of a range of types of storage device and media including magnetic, optical and solid state
- describe how these principles are applied to currently available storage solutions, such as SSDs, HDDs, USB flash memory, DVDs, CDs and Blu-ray discs
- calculate the storage requirement of a file

### 4.3 Magnetic storage

### 4.3 Optical storage

### 4.3 Solid state (flash) storage

### 4.3 Measuring storage

### 4.3 File sizes

#### 1.3.6 Operating systems

- describe the purpose of an operating system (understand the purpose and function of an operating system and why it is needed)
- show understanding of the need for interrupts

#### 1.3.7 High- and low-level languages and translators

- show understanding of the need for both high-level and low-level languages
- show understanding of the need for compilers when translating programs written in a high-level language
- show understanding of use of interpreters with high-level language
- show understanding of the need for assemblers

#### 1.4 Security

##### 1.4.1

- show understanding of the need to keep data safe from accidental damage, including corruption and human errors
- show understanding of the need to keep data safe from malicious actions, including unauthorised viewing, deleting, copying and corruption

##### 1.4.2

- show understanding of how data are kept safe when stored and transmitted, including:
  - use of passwords, both entered at a keyboard and biometric
  - use of firewalls, both software and hardware, including proxy servers
  - use of security protocols such as secure socket layer (SSL) and transport layer security (TLS)
  - use of symmetric encryption (plain text, cypher text and use of a key) showing understanding that increasing the length of a key increases the strength of the encryption

#### Chapter 5, topic 1

- 5.1 What is software?
- 5.1 Operating systems

#### Chapter 5, topic 2

- 5.2 Low-level languages
- 5.2 High-level languages

#### Chapter 6, topic 1

- 6.1 Data security
- 6.1 Security threats
- 6.1 Malpractice and crime
- 6.1 Online attacks

#### Chapter 6, topic 2

- 6.2 Proof of identity
- 6.2 Firewalls
- 6.2 Security protocols
- 6.2 Encryption
<table>
<thead>
<tr>
<th>1.4.3</th>
<th>show understanding of the need to keep online systems safe from attacks including denial of service attacks, phishing, pharming</th>
<th>6.1 Online attacks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.4.4</td>
<td>describe how this knowledge can be applied to real-life scenarios including, for example online banking, shopping</td>
<td>6.1 Malpractice and crime</td>
</tr>
<tr>
<td>6.1 Online attacks</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.2 Security examples</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>1.5 Ethics</strong></td>
<td><strong>Chapter 7</strong></td>
<td></td>
</tr>
<tr>
<td>• show understanding of computer ethics, including copyright issues and plagiarism</td>
<td>7.1 Copyright</td>
<td></td>
</tr>
<tr>
<td>• distinguish between free software, freeware and shareware</td>
<td>7.1 Free software</td>
<td></td>
</tr>
<tr>
<td>• show understanding of the ethical issues raised by electronic communication and computer systems, including hacking, cracking and production of malware</td>
<td>2.3 Malware and hacking</td>
<td></td>
</tr>
<tr>
<td></td>
<td>6.1 Data security</td>
<td></td>
</tr>
<tr>
<td></td>
<td>6.1 Malpractice and crime</td>
<td></td>
</tr>
<tr>
<td></td>
<td>6.1 Online attacks</td>
<td></td>
</tr>
<tr>
<td></td>
<td>7.1 Hackers and crackers</td>
<td></td>
</tr>
<tr>
<td><strong>2.1 Algorithm design and problem solving</strong></td>
<td><strong>Chapter 9</strong></td>
<td></td>
</tr>
<tr>
<td>2.1.1 Problem-solving and design</td>
<td>9.4 Top-down programming</td>
<td></td>
</tr>
<tr>
<td></td>
<td>9.4 Structure diagrams</td>
<td></td>
</tr>
<tr>
<td>• show understanding that every computer system is made up of sub-systems, which in turn are made up of further sub-systems</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>9.4 Code libraries</td>
<td></td>
</tr>
<tr>
<td>• use top-down design, structure diagrams</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• flowcharts, pseudocode</td>
<td>9.3 Analyse algorithms</td>
<td></td>
</tr>
<tr>
<td>• library routines and subroutines</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• work out the purpose of a given algorithm</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• explain standard methods of solution</td>
<td>9.2 Test data</td>
<td></td>
</tr>
<tr>
<td></td>
<td>9.3 Trace tables</td>
<td></td>
</tr>
<tr>
<td>• suggest and apply suitable test data</td>
<td>9.1 Count how many</td>
<td></td>
</tr>
<tr>
<td></td>
<td>9.1 Calculate a total</td>
<td></td>
</tr>
<tr>
<td></td>
<td>9.1 Calculate an average</td>
<td></td>
</tr>
<tr>
<td></td>
<td>9.1 Verification</td>
<td></td>
</tr>
<tr>
<td></td>
<td>9.1 Validation</td>
<td></td>
</tr>
<tr>
<td>• understand the need for validation and verification checks to be made on input data (validation could include range checks, length checks, type checks and check digits)</td>
<td>9.1 Verification</td>
<td></td>
</tr>
<tr>
<td></td>
<td>9.1 Validation</td>
<td></td>
</tr>
<tr>
<td>• use trace tables to find the value of variables at each step in an algorithm</td>
<td>9.3 Trace tables</td>
<td></td>
</tr>
<tr>
<td></td>
<td>9.3 Trace tables (loops)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>9.3 Trace tables (flowcharts)</td>
<td></td>
</tr>
<tr>
<td>• identify errors in given algorithms and suggest ways of removing these errors</td>
<td>9.3 Find errors in algorithms</td>
<td></td>
</tr>
<tr>
<td>• produce an algorithm for a given problem (either in the form of pseudocode or flowchart)</td>
<td>9.3 Create an algorithm</td>
<td></td>
</tr>
</tbody>
</table>
- comment on the effectiveness of a given solution

### 2.1.2 Pseudocode and flowcharts

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
</table>
|  | 9.2 Evaluation  
9.3 Find errors in algorithms |
|  | **Chapter 8** |
|  | 8.2 Variables in pseudocode  
8.2 Variables in flowcharts |
|  | 8.3 Selection in pseudocode  
8.3 Selection in pseudocode |
|  | 8.4 Loops in pseudocode |
|  | 8.2 Sequence  
8.2 Variables in pseudocode |
|  | 9.1 Calculate a total  
9.1 Count how many |
|  | 9.1 Calculate an average  
9.1 Verification  
9.1 Validation |
|  | 8.2 Input  
8.2 Assign a value  
8.2 Calculated values |
|  | 8.3 Logical decision |
|  | 8.4 Loops  
9.1 Count how many  
9.1 Calculate a total  
9.1 Calculate an average  
9.1 Verification  
9.1 Validation |
|  | 8.2 Assign a value |
|  | 8.3 Python if ...  
8.3 Python if ... else ...  
8.3 Python elif |
|  | 8.4 Loops  
8.4 for loops  
8.4 while loops |
|  | 9.1 Calculate a total  
9.1 Calculate an average |

### 2.2 Programming

- **Chapter 8, topics 2 – 4**  
**Chapter 9, topic 1**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
</table>
|  | 8.2 Input  
8.2 Assign a value  
8.2 Calculated values |
|  | 8.3 Logical decision  
8.4 Loops  
9.1 Count how many  
9.1 Calculate a total  
9.1 Calculate an average  
9.1 Verification  
9.1 Validation |
|  | 8.2 Assign a value |
|  | 8.3 Python if ...  
8.3 Python if ... else ...  
8.3 Python elif |
|  | 8.4 Loops  
8.4 for loops  
8.4 while loops |
|  | 9.1 Calculate a total  
9.1 Calculate an average |

### 2.2.1 Programming concepts

- declare and use variables and constants  
- understand and use basic data types: integer, real, char, string and Boolean  
- understand and use the concepts of sequence  
- selection  
- repetition  
- totalling
Using Python

The practical programming lessons in this book are based on the Python programming language. Python is a simple to use programming language which can be downloaded for free and used on any computer without restriction. There are many published books and online resources to help you learn Python and apply it to your teaching.

At a minimum, go through the exercises and activities in this book and you will develop enough confidence and skills to lead your students through the fundamentals of this programming language.

Download a copy
To download a copy of Python go to: https://www.python.org

You will see links to the area of the site where you can download the files you need to write and run Python programs. At the time of writing the URL for this area was: https://www.python.org/downloads/

You or your school or college technician can also use this link to download and install Python on all classroom computers.

| Counting | 9.1 Count how many
| Predefined procedures | 8.2 Output
| Arrays | 9.4 Code libraries

2.2.2 Data structures; arrays
- declare and use one-dimensional arrays
- show understanding of the use of one-dimensional arrays, including the use of a variable as an index in an array
- read or write values in an array using a FOR ... TO ... NEXT loop

Chapter 8, topic 5

Chapter 8, topic 5
- define a single-table database from given data storage requirements
- choose and specify suitable data types
- choose a suitable primary key for a database table
- perform a query-by-example from given search criteria

Chapter 10
- 10.1 Records and fields
- 10.1 Data types
- 10.1 Primary key
- 10.2 Select fields
- 10.2 Select records

Versions
More than one version of Python is available on https://www.python.org. The version numbers are constantly being updated. However, differences between the versions are minor, and it will not matter which version you use. It is recommended you download the most up-to-date version available on the site.

NOTE: There are significant differences between versions of Python which begin with a 2 (e.g. Python 2.3.1.) and those which begin with a 3 (e.g. Python 3.4.4.). This book is written for any version of Python which begins with a 3.

Finding support
On the same website you will find tutorials and documentation. There are discussion forums online for educators who use Python in the classroom. There are worked examples and demonstrations on YouTube. In short, there is a wealth of interesting materials for teachers. Enthusiastic students who want to take their programming further will find plenty of support and discussion online.
Kerboodle

The Complete Computer Science for Cambridge IGCSE and O Level Kerboodle is an online learning platform intended to support your teaching and help your students achieve their best in the course and examination. It provides hundreds of engaging, interactive resources and assessments, while also being closely matched to the student book and its structure.

Use the search bar for a quick and easy way of finding the resources you need.

There are five main modules in this Kerboodle: Lessons, Resources, Assessments, Markbook, Digital Books. You can navigate between them by clicking on the named tabs at the top right of your screen.

To help you find the resources you need, the content follows the student book structure: click on the chapter name, to display all the resources tagged to that chapter; click on a topic name, to only see the resources relevant to that sub-section of the student book.

Lessons

In the Lessons module you will find ready-to-play lesson presentations, complementing each spread in the student book. Each lesson starts by stating the learning objectives, then goes through all the material on the spread and concludes with ideas for assessment.

All the lessons are fully customizable, allowing you to adapt them and include some of your own resources and notes.

Resources are built into the lesson player, helping you make the most of the resources found on Kerboodle.

The lesson player links to the Assessments module, helping you find quizzes, tests and homework you can assign.

The Lessons module is a teacher-only area.
Resources

The Resources module is where you will find hundreds of interactive activities, worksheets, and many other materials to help your students practise and develop their knowledge and skills. All the resources available fall into one of the two categories:

1. Topic-based resources are research tasks and exam support and practice activities that map to each topic of the student book.
2. Lesson-based resources are activities and worksheets that map to each spread of the student book.

On Your Marks

The interactive On Your Marks are designed to help your student build their skills and confidence answering examination questions. Covering a range of question types, each activity is split into two separate parts, allowing for a step-by-step approach:

- **Understand and prepare** (found in the Resources module of Kerboodle) – allows students to analyse an examination question and three sample student answers. It then asks students to look at examiner feedback for those answers and rank them, by giving each a mark.

- **Test** (found in the Assessment module of Kerboodle, only available to students once you assign it to them) – allows students to answer the question they have analysed in the Understand and prepare part for themselves.

You will be notified once your students have answered the On Your Marks: Test part you assigned to them; you will then need to manually mark their work. To help with this, a mark scheme for each of the questions is provided.

WebQuests

The WebQuest mini projects allow your students to research specific aspects of the topics covered in the student book. These projects are designed to strengthen their understanding of the subject and help them develop skills such as communication and teamwork.

4.3 Using storage devices and media

Introduction

Storage devices form a category of computer hardware. The WebQuest is concerned with secondary storage devices, which are the ones that allow you to make more permanent copies of your files and data. Each type of storage device has different strengths and weaknesses, which make it more suitable for certain purposes.

Task

1. Do some research to find out about a range of storage devices and media. Some examples are hard disk drives (fixed and removable), solid state drives, DVD, CD, Blu-ray disc and USB flash memory. Your teacher may ask each group to research a different category of storage device.

2. First, find out how each device works, and if appropriate, what storage medium it uses. Try to discover the main benefits and drawbacks of each type of storage device and medium. You will then describe how each device is used in real life scenarios.

3. Create a presentation of about five minutes' duration for the rest of your class. Your presentation should be informative, concise and include pictures of the devices and, if relevant, their associated media.

Process

1. **Step 1: Roles**

   A useful way to organise this task would be for each group to be allocated a category of storage device and medium to research. Your group should decide how to tackle the task you have been assigned. One method would be for each member of the group to initially look at a particular type of storage device and medium, and then gather as much useful information as possible. Once this is done, the group as a whole will create the final presentation.

2. **Step 2: Research**

   Carry out the research on your specific storage device(s) and any associated media, and try to discover as much as you can about how it works and its good and bad points. Make sure you have a picture of it. Then find out and explain how and where it can be used in real life scenarios.
Interactive starter activities*

There is a starter activity for each spread of the student book, specifically designed to consolidate your students’ understanding of key concepts in computer science.

*A copy of each of the starter activities can be found in the Assessments module, allowing you to assign them as homework.

Extension worksheets

There is an extension worksheet for each spread of the student book; these worksheets are designed to extend your students’ understanding of the concepts covered in that lesson.

Java and Visual Basic programming support

The student book uses Python as the example programming language. However, if you are teaching Visual Basic or Java, these PDF booklets will bring you all the material on the programming spreads in the student book (chapters 8 and 9), but with specific Java or Visual Basic syntax and examples.
Assessments

In the Assessments module, you will find homework, quizzes, tests and exam-style practice papers. By default, these are hidden in the student view. For students to be able to see them, they need to be assigned. Once assigned, assessments can be tracked in the Markbook and you can run reports on your students’ progress.

End of topic quizzes
The multiple-choice quizzes allow you to test your students’ knowledge at the end of each topic and identify any areas of improvement. They are auto-marked and offer post-submission summative feedback.

Starter activities (homework)
A copy of the interactive starter activities from the Resources module; they are auto-marked and offer formative feedback, allowing students to attempt each question three times before submitting it.

Exam-style practice papers
The exam-style practice papers closely follow the structure of Paper 1 (Theory of Computer Science) and Paper 2 (Problem-solving and programming skills), respectively. There are three practice papers for each of the two exams, accompanied by detailed mark schemes and pre-release materials for Paper 2.
Digital books
In this module you will find a digital copy of the student book, as well as a copy of the teacher guide. You can use the student book to navigate through the resources in the Resources module of Kerboodle, as they are all tagged to the relevant spread. You can also display these books, annotate and personalise them.

_The Digital Books module is a teacher-only area._

Answers and mark schemes
To help you check and mark your students’ work, the following files are available on Kerboodle:

- Answer files to the extension worksheets in the Resources module
- Answer files to the ‘Test yourself’ questions in the student book (also posted on the free website www.oxfordsecondary.com/9780198367215)
- Mark schemes for On Your Marks: Test
- Mark schemes for exam-style practice papers
1.1 Binary data

Resources
1.1 Binary data: lesson
1.1 Binary data: activity
1.1 Binary data: worksheet

Objectives
By the end of the lesson students should be able to:

• explain what a computer does
• explain why a computer uses binary data
• (extension) assess extended ability by identifying different types of information and identifying binary numbers

Overview
In this lesson students will be introduced to the most basic concepts in computing. This lesson introduces the subject of binary data and binary numbers. Students will learn that computers process data to make information and that this processing occurs in digital form.

Before the lesson

• Obtain copies of the Student Book for the class.
• Make sure all students have access to a copy of the course timetable.
• If you school has a subscription to the Complete Computer Science for Cambridge IGCSE and O Level Kerboodle, make sure all students have access to the Resources module on the platform.
• A copy of the extension worksheet will be needed to support any student ready to proceed to extension activities. (This needs to be kept in mind for all subsequent lessons.)
• (Optional) Find examples of suitable and age-appropriate websites that students can look through to find examples of data. As the Internet is always changing, it is important to check these immediately before recommending them to students.

Activities

Teacher-centred learning

• Give an overview of the course content:
  – distribute and/or introduce the Student Book and explain ways of working
  – give out the course schedule.

• Point out that there will be two lessons each week and each lesson corresponds to a double-page spread of the Student Book.
• Introduce the lesson: the fundamental ideas of computer science (data processing and binary).
• Present the basic diagram of data processing. Invite students to suggest examples of data being processed to create information from computer systems. Students should draw the diagram that appears in the Student Book and make notes of the examples.
• Present the concept of binary states and how these are represented using binary digits.

Student activities

• Students make notes as the teacher presents the diagram and major concepts.
• Working in groups, students develop answers to the “Test yourself” questions.
• In a class discussion, review the answers that students have given. In particular, explore the answer to question 4. Students amend their written answers during the discussion, to note extra points raised by the teacher and other students.

Individual learning activity:

• Students carry out research using the Internet to find samples of the types of data mentioned in class.

Assessment of learning:

• Use the interactive starter activity 1.1 Binary data: complete sentences with missing words to demonstrate understanding.

Extension

More-able students will develop more complex answers to question 4 of the “Test yourself” section. Use the extension worksheet 1.1 Binary data: assess extended ability by identifying different types of information and identifying binary numbers.

Homework

Students type up full answers to the “Test yourself” questions using word-processing software. They save their work to their storage area.

1.1 Binary data extension worksheet or 1.1 Bits and bytes starter activity may be given as homework. Students use graphics software to create a version of the data-processing diagram drawn in class.
1.1 Bits and bytes

Resources
1.1 Bits and bytes: lesson
1.1 Bits and bytes: activity
1.1 Bits and bytes: worksheet

Objectives
By the end of the lesson students should be able to:
- explain what a byte is
- measure computer memory in terms of the number of bytes of data it can hold
- (extension) explain memory factors that increase the speed and power of computer systems.

Overview
Students have learned that data in the computer is held in binary digital form. In this lesson they will learn how binary data is stored in computer memory. They will understand that computer memory is measured in terms of how much data it will hold and that larger memory is linked to processing speed.

Before the lesson
- Make sure the relevant materials are available for students.
- (Optional) Find either magazines with computer adverts or tested and trusted website links where students can research specifications (including RAM size) of computers for sale.

Activities
Teacher-centred learning
- Review what students have learned about the significance of binary data to computer systems.
- Introduce the lesson: explain the meaning of bit and byte. Describe the units of computer memory (kilobyte, megabyte, and gigabyte).
- (Extension) Discuss registers and factors that make computers work faster.

Student activities
- Students make notes as the teacher presents learning content.
- Students write answers to the “Test yourself” questions. Question 4, in particular, is an extension task. Students mark and correct their own answers.
- In a class discussion on the speed of computers, ask: Which applications require speed of processing? How does expanding RAM help with this?
- Students carry out an Internet search (or search magazines) for adverts for personal computers, mentioning the size of RAM. Collate findings.

Assessment of learning:
- Use the 1.1 Bits and bytes interactive starter activity: put units of memory in order; calculate storage size.

Extension
More-able students will develop longer answers to “Test yourself” question 4.

Use the extension worksheet 1.1 Bits and bytes: calculate storage requirements for example files.

Homework
Students write a full answer to test question 4, explaining how upgrading RAM increases the speed of a computer. Explain all the terms used.
1.1 Binary and denary

Resources
1.1 Binary and denary: lesson
1.1 Binary and denary: activity
1.1 Binary and denary: worksheet

Objectives
By the end of the lesson students should be able to:

- explain what binary and denary numbers are
- recognise binary numbers and the components of these numbers
- explain the difference between a digit and a number
- explain how positional values change the significance of digits in numbers
- (extension) explain the history and development of denary and binary number systems.

Overview
Students have learned that the computer holds data in binary digital form. In this lesson they will learn the difference between binary and denary number systems. This requires them to understand the difference between a digit and a number, and how the positional value of a digit changes its significance within a number (in any number system).

Before the lesson
- Make sure the relevant materials are available for students.
- (Optional) Investigate websites with relevant information about the development of number systems through history, which will help you to direct the web research of more-able students.

Activities
Teacher-centred learning
- Remind students of the significance of binary and the meanings of bit and byte.
- Introduce the lesson: students will learn how to interpret binary numbers.
- Describe the denary number system, and the positional value of digits based on powers of ten.

Student activities
- Students make notes as the teacher presents the key learning content.
- They write answers to the “Test yourself” questions working independently, then mark their own work during a class discussion.
- Students participate in a class discussion on the history of number systems, and the contribution of different societies in the past including Indian, Persian and Arabic mathematicians.

Assessment of learning:
- Use the interactive starter activity 1.1 Binary and denary: convert binary numbers to denary using positional values.

Extension
More-able students can search the Internet for information about the history and development of number systems.

Homework
Students complete extension worksheet 1.1 Binary and denary: complete a crossword by putting the names of denary numbers into a grid.
1.1 Counting in binary

Resources
1.1 Counting in binary: lesson
1.1 Counting in binary: activity
1.1 Counting in binary: worksheet

Objectives
By the end of the lesson students should be able to:
• count up in binary
• understand the numerical significance of binary numbers in the context of a counted sequence
• (extension) develop binary numbers using positional values.

Overview
Students have learned the difference between binary and denary number systems. In this lesson they will develop their understanding of what binary numbers mean, by learning how to count in binary from zero up.

Before the lesson
• Make sure the relevant materials are available for students.
• Make sure you understand the group activity and how it will be implemented in class. You may need to rearrange the classroom so there is a row of eight chairs and plenty of space to use for the activity.

Activities
Teacher-centred learning
• Remind students of what they have learned so far about the meaning of binary numbers.
• Introduce the lesson: students will learn how to count in binary.
• Describe counting in binary. Count in binary on the whiteboard.

Student activities
• Students make notes on counting in binary.
• Students enact binary counting through a physical activity. Individual students stand for the “columns” of a binary number. The activity is described in full in the Student Book.
• Students review the “Learning activity” in a discussion: What was the activity like and has it helped them to understand binary numbers?

Assessment of learning:
• Use 1.1 Counting in binary interactive starter activity: convert binary numbers to denary using positional values.

Extension
More-able students will be able to take the lead in the class activity. Once they have practised with assistance they will be able to work without the teacher’s instruction. This is an opportunity for them to lead or organise other students.

Use the extension worksheet 1.1 Counting in binary: work with a partner to develop binary numbers using positional values.

Homework
Students write up the “Learning activity” they did as a class, and explain from this how to count in binary.
1.1 Convert binary to denary

Resources
1.1 Convert binary to denary: lesson
1.1 Convert binary to denary: activity
1.1 Convert binary to denary: worksheet

Objectives
By the end of the lesson students should be able to:
• convert binary numbers to their denary equivalent
• (extension) explain the process of binary to denary conversion.

Overview
Students have learned what binary numbers are and how to count in binary. In this lesson they will learn a defined mathematical skill: the conversion of binary numbers to the denary equivalent.

Before the lesson
• Make sure the relevant materials are available for students.
• (Optional) You may wish to make Microsoft PowerPoint or similar presentation software available for students to use for the activities.

Activities
Teacher-centred learning:
• Review what students have learned about counting in binary, and about the significance of position in determining the value of a binary digit (bit).
• Introduce the lesson: students will convert binary numbers to denary (remind students that denary is also called decimal).
• Demonstrate the conversion of binary to denary on the whiteboard.

Student activities
• Students make notes on the conversion of binary to denary.
• Students write answers to the “Test yourself” questions to demonstrate understanding. They check and mark answers as a class activity.
• Students work in pairs or small groups to draft an explanation of the binary to denary conversion process. Microsoft PowerPoint or similar software could be used to prepare materials as if for a lesson. If there is time, students will share these explanations or deliver the lesson with other groups to check understanding.

Assessment of learning:
• Use the 1.1 Convert binary to denary interactive starter activity: carry out a range of number conversions.

Extension
More-able students will be able to articulate the process of conversion between denary and binary. Use the extension worksheet 1.1 Convert binary to denary: use binary to denary conversion to complete a puzzle game.

Homework
In the “Learning activity” students write a handout or other material to explain to young learners how to convert 8-bit binary numbers into denary.
1.1 Convert denary to binary

Resources
1.1 Convert denary to binary: lesson
1.1 Convert denary to binary: activity
1.1 Convert denary to binary: worksheet

Objectives
By the end of the lesson students should be able to:
• convert denary number to their binary equivalent
• complete the summative (end-of-topic) assessment on binary numbers
• (extension) explain the process of denary to binary conversion.

Overview
Students have learned how to convert binary to denary. In this lesson they will learn how to convert denary to binary.

Before the lesson
• Make sure the relevant materials are available for students.

Activities
Teacher-centred learning
• Review what students have learned about converting binary to denary numbers and using “position values” of the bits.
• Introduce the lesson: students will convert denary to binary, by reversing the process they have learned.
• Demonstrate the conversion of denary to binary on the whiteboard.

Student activities
• Students make notes on denary to binary conversion.
• Students write answers to the “Test yourself” questions to demonstrate understanding. They check and mark answers as a class activity.
• Students work in pairs or small groups to draft an explanation of the conversion process. If there is time, students will share these explanations with other groups to check understanding. As with the previous class, students may want to use Microsoft PowerPoint and develop the work into a “lesson”.

Assessment of learning:
• Use the 1.1 Convert denary to binary interactive starter activity: carry out a range of number conversions.
• Use the summative quiz to assess learning over the complete topic.

Extension
More-able students will be able to articulate the process of conversion between denary and binary. Use the extension worksheet 1.1 Convert denary to binary: use denary to binary conversion to complete a puzzle game.

Homework
“Learning activity”: students write a handout or other material to explain to young learners how to convert denary to binary.

End of topic resources
The six lessons in the 1.1 Binary systems topic provide learning opportunities in relation to binary systems. The following resources were designed to extend and test your students’ knowledge of this topic:
• 1.1 Binary systems: Representing numbers in binary WebQuest
• The two-part On Your Marks interactive activity: 1.1 Binary systems: Understand and prepare and 1.1 Binary systems: Test, as well as a mark scheme for the test
• The ten-question 1.1 Binary systems quiz, a multiple choice test. Students should be aiming to achieve a mark of 80 per cent or more. If students achieve a low mark for this topic, they need to go back and revise the material.