

Published for
OXFORD INTERNATIONAL
AQA EXAMINATIONS

International GCSE **COMPUTER SCIENCE**

Alison Page

OXFORD

Contents

1 Foundations of programming

1.1 Begin programming

Introduction to Python	2
Output	4
Variables	6
Input	8
Flowcharts	10
Pseudocode	12

1.2 Calculation

Numerical expressions	14
Data types	16
Convert data type	18
Integer division	20
Constants	22
Processing values	24

1.3 String handling

Strings	26
Substrings	28

1.4 Selection

Selection	32
if... else...	34
Nested if	36
Boolean operators	38
Selection in pseudocode	40

1.5 Repetition

Loops	42
Counter loops	44
Condition loops	46
Nested loops	48
Loops in pseudocode	50
End-conditional	52

2 Further programming

2.1 Data structure

Lists	54
Output a list	56
Add elements to a list	58
Arrays in pseudocode	60
2-D arrays and lists	62
Records	64
Make a list of records	66

2.2 File handling

Write to a file	68
Read from a file	70
Files in pseudocode	72

2.3 Subroutines

Introduction to subroutines	74
Random numbers	76

Define and call procedures	78
Parameters	80
Multiple parameters	82
Functions and returned values	84

2.4 Structured programming

The structured approach to programming	86
Advantages of the structured approach	88
Structured algorithms	90

2.5 Robust programming

Data validation	92
Secure programming	94
Testing	96

3 Representing algorithms

3.1 Interpret algorithms

Algorithms and programs	98
Abstraction	100
Purpose of an algorithm	102
Trace tables	104
Trace tables with loops	106

3.2 Search algorithms

Linear search	108
Search function	110
Search a sorted list	112
Implement the binary search algorithm	114
Compare search algorithms	116

3.3 Sort algorithms

Sorting a list	118
Bubble sort	120
Merge sort	122
Compare sort algorithms	124
Graphing time efficiency	126

4 Number systems

4.1 Binary systems

Binary data	130
Bits and bytes	132
Binary and decimal	134
Counting in binary	136
Convert between binary and decimal	138

4.2 Hexadecimal

What is hexadecimal?	140
Hexadecimal and binary	142
How hexadecimal is used	144

4.3 Binary arithmetic

Binary addition	146
Binary multiplication	148
Binary division	150

5 Digital data

5.1 Character encoding

Character codes	152
Unicode	154
Converting text to characters	156

5.2 Images

Bitmaps and pixels	158
Image quality and file size	160
Turn an image into binary	162
Creating coloured images	164

5.3 Sound

Analogue sound	166
Calculating file size	168

5.4 Compression

Why compression is important	170
Run-length encoding	172
Huffman coding	174
Build a Huffman tree	176
Huffman compression	178

6 Software

6.1 Types of software

Hardware and software	182
System and application software	184
Operating systems and utilities	186

6.2 Programming languages

Machine code and assembly language	188
Low-level and high-level languages	190
Translators	192

7 Hardware

7.1 The processor

Components of the CPU	194
The fetch-execute cycle	196
Main memory	198
Performance factors	200
Embedded systems	202

7.2 Storage

Magnetic storage	204
Optical storage	206
Solid state storage	208
Choosing the right storage	210

7.3 Boolean logic

Logic gates and circuits	212
NOT gate	214
AND gate	216
OR gate	218
Logic circuits	220
Interpret logic circuits	222
Logic circuits and Boolean expressions	224

8 Communications

8.1 Networks

What is a network?	226
Types of network	228
Network topologies	230

8.2 Protocols

Network protocols	232
The four-layer model	234
The four-layer model (continued)	236

9 Security

9.1 Security threats

Importance of security	238
Threats to security	240
Social engineering	242
Malware	244

9.2 Ensuring security

Cyber-security protection	246
Network security	248
Encryption	250
Penetration testing	252

10 Databases

10.1 Database design

Introducing databases	254
Table design	256
Primary key	258
Relational databases	260
Database design	262

10.2 Structured query language (SQL)

Add data with SQL	264
Edit data with SQL	266
Select data with SQL	268
Select data from two tables	270

11 Web pages

11.1 How webpages are made

What is Hypertext Mark-up Language (HTML)?	272
Designing web pages with cascading style sheets (CSS)	274
Client-side and server-side scripting	276

11.2 Creating web content

Building a basic web page	278
Using inline and block-level tags to improve web page layout	280
Selections, images and links	282
Designing a web page using CSS	284
Answers	286
Index	305

Introduction

If you are studying Computer Science for the Oxford AQA International GCSE, then this book is designed for you. Its purpose is to help you achieve your best in the course and examination, equipping you with the knowledge you need to study the subject at a higher level.

The book matches the syllabus exactly and consists of eleven chapters: chapters 1-3 cover exam paper 1 of the syllabus (**Programming**), while chapters 4-11 cover exam paper 2 (**Concepts and Principles of Computer Science**).

To help you make the most of this student book, the following features are used to organise the content:

Syllabus reference

The explicit syllabus reference allows you to frame your learning and build connections between different topics.

Introduction

The short introduction will make it clear what you will learn in each lesson.

Example

Examples are given to show you what your screen should look like or to give step-by-step guidance on what to do.

Q

Test yourself & Learning activity

The “Test yourself” and “Learning activity” questions will help you check your understanding after each lesson and give you further opportunities to practise what you have learnt.

Extension activity

The Extension Activities go beyond the requirements of the syllabus and will not be tested in the examination. They are intended to give you a broader understanding of computer science and hopefully you’ll find them good practice.

Programming

In this student book, Python is used as an example programming language. There are many other programming languages and your teacher might have chosen a different one. However, the features you will learn about are found in almost all languages.

Syllabus reference

3.2.2 Programming concepts

Students should be able to write programs.

1.1 Begin programming

Introduction to Python

Introduction

This book will help you to learn about Computer Science. The first thing you will learn is how to write computer programs. As you learn to write programs, you will learn more about how a computer works.

You will learn to write programs in the Python programming language.

Begin programming

In this chapter you will learn how to write computer programs. The Python programming language is used as an example. There are many other programming languages. The features you will learn about are found in almost all languages. You will write Python programs that use variables, loops, and other key features. Programs written in other languages also have these features.

To extend your understanding of programming you could work independently to learn about an additional language and see how it compares to Python. This is not essential.

What is Python?

Python is a programming language. You can write programs in the Python language. Then you can run the programs. You will control the actions of the computer.

You do not have to pay to use Python. You can download Python onto your own computer at home. That means you can practise programming outside of classroom time. To download Python go to the main Python website:

<https://www.python.org/>

Choose “Downloads” from the menu.

Versions

There are several versions of Python. Each version has a different number. Some start with 2, for example 2.7.10. Some start with 3, for example 3.5.3. In this book we use version 3. You can download and use any version as long as it begins with 3.

Integrated development environment (IDE)

Like many programming languages, Python comes with an integrated development environment (IDE). The Python IDE is called IDLE. An IDE is similar to a word processor. An IDE lets you type up your program and save it like a document. However, a typical IDE has extra features you don't find in a word processor. For example, IDLE:

- uses colour to show different features of the program code

This code assigns the value 23 to the variable Age:

```
Age = 23
```

This code assigns the value "Bond" to the variable CodeName:

```
CodeName = "Bond"
```

Declaring variables

In some programming languages you must create a variable with a special command before you can use it. When you create the variable you give it an identifier and a data type (find out more about data types on page 16). The variable is empty – it doesn't have a value. Creating a variable is called 'declaring' or 'initialising' the variable. In Python, you do not need to declare variables using a special command. Just start using the variable when you need it in your program. A variable is declared when you assign a value to it for the first time.

Output a variable

You have learned that a print command will output a value. You can put the name of a variable into the print command. The computer will output the value stored in the variable:

```
print(Age)
print(CodeName)
```

When you output a variable you do not put quote marks round the variable name. You don't want the computer to print out the name of the variable as a word. You want it to print the value that is stored in the variable.

You can combine a string and a variable in the same output command. Put a comma between them:

```
print("Your code name is ", CodeName)
```

Test yourself

1. What is the name for code that represents a data value?
2. Give the command to assign the value 19.99 to the variable TicketCost.
3. Give the command to output the value stored in the variable TicketCost.
4. Here are four variable names. For each one say whether it is a good or bad variable name and say why.
 - a Character Name
 - b firstName
 - c 5Star
 - d myvariable
 - e Star*Rating

Learning activity

Open the computer game program you started writing in the last lesson.

1. Create a variable called Score.
2. Assign the value 0 to this variable.
3. Add a command to print out the variable.

Syllabus reference

3.3.5 Character encoding

Understand what a character set is and be able to describe this character encoding method: 7-bit ASCII.

Understand that character codes are commonly grouped and run in sequence within encoding tables.

Additional information

Students should be able to use a given character encoding table to: convert characters to character codes, convert character codes to characters.

3.3.8 Data compression

Be able to calculate the number of bits required to store a piece of uncompressed data in ASCII.

5.1 Character encoding

Character codes

Introduction

Computers work with all types of data, including numbers, text, images and sound. All these different types of data must be held in binary form inside the computer. When data is held in binary form it is called digital data. In this lesson you will learn how text characters are stored using a code called ASCII.

Digital data

“Digital” is used to refer to a value that can be represented in digits as a number of units. A digital value does not vary in a smooth curve, but can be represented by a series of distinct values. All data held inside a computer is digital because the values are held as a pattern of on/off switches. We can represent these values as binary numbers.

Text data

The computer stores text using a number code system. Every text character is given a code number. The first widely used system of text coding was called ASCII. ASCII stands for American Standard Code for Information Interchange. The first version of ASCII was produced in the USA in 1963.

ASCII has a code number for each of the common text characters on the standard American computer keyboard. There are different codes for upper-case and lower-case letters. There are codes for the other characters that can be typed on the keyboard such as punctuation.

ASCII codes

The ASCII code system is sequential. That means the number codes go up in number order. ASCII number codes can be represented as decimal, binary or hexadecimal values. The table shows the ASCII codes for some of the upper-case letters of the alphabet. The code numbers are shown in decimal, binary and hexadecimal.

Character	Decimal	Binary	Hexadecimal
A	65	100 0001	41
B	66	100 0010	42
C	67	100 0011	43
D	68	100 0100	44
E	69	100 0101	45
F	70	100 0110	46
G	71	100 0111	47
H	72	100 1000	48
I	73	100 1001	49

Syllabus reference

3.7.3 Structured query language (SQL)

Use the SQL commands `UPDATE` and `DELETE FROM` to edit and delete data in a database.

Edit data with SQL

Introduction

You have learned to add data to a SQL table using the `INSERT` command. In this section you will learn to delete records from a table. You will learn to make changes to the values in a table. These actions can be carried out with SQL commands.

The `DELETE FROM` command

You can delete all the records from a data table. Then you will be left with nothing but an empty table. For example, at the start of a new football season, the match records from the previous season might be deleted.

The command to delete all records is `DELETE FROM`. Complete the command by giving the name of the table:

```
DELETE FROM Match
DELETE FROM Player
```

The `WHERE` command

SQL has a command that lets us pick out one or more records from the table. It is the `WHERE` command. This command is always followed by a logical test. The computer will pick the records where the result of the logical test is True.

Here is a general example:

```
WHERE field = value
```

Look at the table below. It contains data about replacement hard drives for computers.

Product Code	Product Name	Supplier	Description	Cost
HD001	Canvio Basics 1TB hard drive	Toshiba	Portable External Hard Drive 2.5 Inch USB 3.0 - Black	47.98
HD002	Maxtor M3 1 TB hard drive	Seagate	USB 3.0 Slimline Portable Hard Drive - Black	49.23
HD003	M3 Slimline 2 TB hard drive	Samsung	USB 3.0 Portable Hard Drive - Black	207.00

Here is an example of a `WHERE` command:

```
WHERE Supplier = "Seagate"
```

This tells the computer to find records `WHERE` the supplier is Seagate. We must put the value "Seagate" in quote marks because it is a text field.

The computer finds this record.

Product Code	Product Name	Supplier	Description	Cost
HD002	Maxtor M3 1 TB hard drive	Seagate	USB 3.0 Slimline Portable Hard Drive - Black	49.23