Heathcote School & Science College

Computer Science Department

OCR A Level GCE
Computer Science

Transition Guide

2018 - 2019

Name:	
Teacher/s:	

Heathcote School and Science College

Course Introduction

Welcome to the OCR GCE Computer Science guide. This qualification is a two year A level course, which prepares you for employment or higher education.

This course aims to develop your skills, knowledge and understanding of the sector, and prepare you for further study or training.

The course is assessed through examinations & a programming project which is worth 20% of total A level.

The purpose of the handbook is to guide you through the course and answer any questions that you may have.

How the course will be run

The OCR GCE Computer Science course will be run by the ICT Department. The

role of your teacher:

- Monitor your attendance and punctuality in lessons
- Monitor your organisation on the course
- Provide you with information regarding the course from OCR
- Check on your academic progress at all stages and units
- Liaise regularly within the department, with the moderator and, where necessary your parent(s)/guardian(s)
- Provide references to prospective employers and other academic institutions when requested
- Provide you with information relating to all of the units

You will have **6 hours of lessons per week**. You will also need to schedule your non-class hours for research, planning and completing coursework tasks.

Content Overview	Assessment Overview	
• The characteristics of contemporary processors, input, output and storage devices • Software and software development • Exchanging data • Data types, data structures and algorithms • Legal, moral, cultural and ethical issues • Elements of computational thinking • Problem solving and programming • Algorithms to solve problems and standard algorithms *The learner will choose a computing problem to work through according to the guidance in the specification.* • Analysis of the problem • Design of the solution • Developing the solution • Evaluation	Computer systems (01) 140 marks 2 hours and 30 minutes written paper (no calculators allowed)	**40%** of total A level
	Algorithms and programming (02*) 140 marks 2 hours and 30 minutes written paper (no calculators allowed)	**40%** of total A level
	Programming project 03* – Repository or 04* – Postal or 80 – Carry forward (2018 onwards)* 70 marks Non-exam assessment	**20%** of total A level

Heathcote School and Science College

1.1 The characteristics of contemporary processors, input, output and storage devices				
Specification Point	Element	Potential	Developing	Secure
1.1.1 Structure and function of the processor	(a)The Arithmetic and Logic Unit; ALU, Control Unit and Registers (Program Counter; PC, Accumulator; ACC, Memory Address Register; MAR, Memory Data Register; MDR, Current Instruction Register; CIR). Buses: data, address and control: how this relates to assembly language programs			
	(b) The Fetch-Decode-Execute Cycle; including its effects on registers.			
	(c) The factors affecting the performance of the CPU: clock speed, number of cores, cache.			
	A-LEVEL ONLY *(d) The use of pipelining in a processor to improve efficiency.*			
	(e) Von Neumann, Harvard and contemporary processor architecture			
1.1.2 Types of processor	(a) The differences between and uses of CISC and RISC processors			
	A-LEVEL ONLY (b) GPUs and their uses (including those not related to graphics).			
	(c) Multicore and Parallel systems.			
1.1.3 Input, output and storage	(a)How different input, output and storage devices can be applied to the solution of different problems.			
	(b)The uses of magnetic, flash and optical storage devices			
	(c) RAM and ROM.			
	(d) Virtual storage.			

Heathcote School and Science College

1.2 Software and software development				
Specification Point	Element	Potential	Developing	Secure
1.2.1 Systems Software	(a) The need for, function and purpose of operating systems.			
	(b) Memory Management (paging, segmentation and virtual memory).			
	(c) Interrupts, the role of interrupts and Interrupt Service Routines (ISR), role within the Fetch-Decode-Execute Cycle.			
	(d) Scheduling: round robin, first come first served, multi-level feedback queues, shortest job first and shortest remaining time			
	(e) Distributed, embedded, multi-tasking, multi-user and Real Time operating systems.			
	(f) BIOS.			
	(g) Device drivers.			
	(h) Virtual machines, any instance where software is used to take on the function of a machine, including executing intermediate code or running an operating system within another.			
1.2.2 Applications Generation	(a) The nature of applications, justifying suitable applications for a specific purpose.			
	(b)Utilities.			
	(c) Open source vs closed source.			
	(d) Translators:Interpreters, compilers and assemblers.			
	A-LEVEL ONLY (e)Stages of compilation (lexical analysis, syntax analysis, code generation and optimisation).			
	A-LEVEL ONLY (f) Linkers and loaders and use of libraries.			
1.2.3 Software Development	(a) Understand the waterfall lifecycle, agile methodologies, extreme programming, the spiral model and rapid application development.			
	(b) The relative merits and drawbacks of different methodologies and when they might be used.			
	(c) Writing and following algorithms.			
A-LEVEL ONLY 1.2.4 Types of Programming Language	(a) Need for and characteristics of a variety of programming paradigms.			
	(b) Procedural languages.			
	(c) Assembly language (including following and writing simple programs with the Little Man Computer instruction set).			
	(d) Modes of addressing memory (immediate, direct, indirect and indexed).			
	(e) Object-oriented languages (see appendix 5d for pseudocode style) with an understanding of classes, objects, methods, attributes, inheritance, encapsulation and polymorphism			

Heathcote School and Science College

AS-LEVEL ONLY - Introduction to Programming				
Specification Point	Element	Potential	Developing	Secure
1.2.3 Introduction to programming	Program Flow			
	Variables and Constants			
	Procedures and Functions			
	Mathematical Operators			
	String handling			
	File handling			
	Assembly Language			

A-LEVEL ONLY - 1.3 Exchanging data				
Specification Point	Element	Potential	Developing	Secure
A-LEVEL ONLY 1.3.1 Compression, Encryption and Hashing	(a) Lossy vs Lossless compression.			
	(b) Run length encoding and dictionary coding for lossless compression.			
	(c) Symmetric and asymmetric encryption.			
	(d) Different uses of hashing.			

Heathcote School and Science College

1.3 Exchanging data				
Specification Point	Element	Potential	Developing	Secure
1.3.2 Databases	(a) Relational database, flat file, primary key, foreign key, secondary key, entity relationship modelling, normalisation and indexing.			
	(b) Methods of capturing, selecting, managing and exchanging data			
	A-LEVEL ONLY (c) Normalisation to 3NF			
	A-LEVEL ONLY (d) SQL – Interpret and modify.			
	A-LEVEL ONLY (e) Referential integrity.			
	A-LEVEL ONLY (f) Transaction processing, ACID (Atomicity, Consistency, Isolation, Durability), record locking and redundancy.			
1.3.3 Networks	(a) Characteristics of networks and the importance of protocols and standards.			
	(b) The internet structure: • The TCP/IP Stack. • DNS • Protocol layering. • LANs and WANs. • Packet and circuit switching.			
	A-LEVEL ONLY (c) Network security and threats, use of firewalls, proxies and encryption.			
	A-LEVEL ONLY (d) Network hardware.			
	(e) Client-server and peer to peer.			
1.3.4 Web Technologies	(a) HTML, CSS and JavaScript.			
	A-LEVEL ONLY (b) Search engine indexing.			
	A-LEVEL ONLY (c) PageRank algorithm.			
	A-LEVEL ONLY (d) Server and client side processing.			

Heathcote School and Science College

1.4 Data types, data structures and algorithms				
Specification Point	Element	Potential	Developing	Secure
1.4.1 Data Types	(a)Primitive data types, integer, real/floating point, character, string and Boolean.			
	(b) Represent positive integers in binary.			
	(c) Use of sign and magnitude and two's complement to represent negative numbers in binary.			
	(d) Addition and subtraction of binary integers.			
	(e) Represent positive integers in hexadecimal.			
	(f) Convert positive integers between binary hexadecimal and denary.			
	(g) Representation and normalisation of floating point numbers in binary			
	A-LEVEL ONLY (h) Floating point arithmetic, positive and negative numbers, addition and subtraction.			
	A-LEVEL ONLY (i) Bitwise manipulation and masks: shifts, combining with AND, OR, and XOR.			
	(j) How character sets (ASCII and UNICODE) are used to represent text.			
1.4.2 Data Structures	(a) Arrays (of up to 3 dimensions), records, lists, tuples			
	A-LEVEL ONLY (b) The following structures to store data: linked-list, graph (directed and undirected), stack, queue, tree, binary search tree, hash table.			
	A-LEVEL ONLY (c) How to create, traverse, add data to and remove data from the data structures mentioned above. (NB this can be either using arrays and procedural programming or an object-oriented approach).			
1.4.3 Boolean Algebra	(a) Define problems using Boolean logic.			
	(b) Manipulate Boolean expressions, including the use of Karnaugh maps to simplify Boolean expressions.			
	A-LEVEL ONLY (c) Use the following rules to derive or simplify statements in Boolean algebra: De Morgan's Laws, distribution, association, commutation, double negation.			
	(d) Using logic gate diagrams and truth tables.			
	A-LEVEL ONLY (e)The logic associated with D type flip flops, half and full adders.			

Heathcote School and Science College

1.5 Legal, moral, cultural and ethical issues				
Specification Point	Element	Potential	Developing	Secure
1.5.1 Computing related legislation	(a)The Data Protection Act 1998			
	(b) The Computer Misuse Act 1990.			
	(c) The Copyright Design and Patents Act 1988.			
	(d) The Regulation of Investigatory Powers Act 2000.			
1.5.2 Moral and ethical Issues	The individual moral, social, ethical and cultural opportunities and risks of digital technology: • Computers in the workforce. • Automated decision making. • Artificial intelligence. • Environmental effects. • Censorship and the Internet. • Monitor behaviour. • Analyse personal information. • Piracy and offensive communications. • Layout, colour paradigms and character sets.			

Heathcote School and Science College

Computing

Components of a computer system

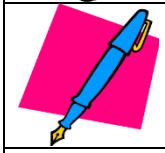
For the exam

- Define the terms hardware, software, input device, storage device and output device;
- Describe the purpose of input devices, storage devices and output devices;
- Describe the different roles and functions of systems software and applications packages.

Introduction

You should use this book alongside the classroom activities to complete. **IT IS NOT** a full and complete guide, in a number of places you will see extension tasks which are designed for higher grades to make sure you thoroughly understand a topic, if you want a grade higher than a B, I suggest you do these.

Symbols used in this book:

	Whenever you see this image it is a dictionary definition or keyword you need for this course
	When you see this image, it requires you to find definitions from the internet, but also describe what this term means in your own words/full sentences.
	When you see this image, this is a class activity that you must complete, usually a diagram and a definition.
	This is an extension task. It is optional, but will help you in the exam

Heathcote School and Science College

Definitions of Key terms

Computing is all about the definitions. If you can name the key terms in Computing you can gain at least one mark in the exam (Hurrah!) In the space below, write out an acceptable definition that you could use in an exam.

Hardware

Software

Input

Output

Storage device

Heathcote School and Science College

Purpose of Hardware

Having these definitions are all well and good, but we also need to be able to describe what the components are for. In the exam, you have to be good at this. In GCSE a short answer will get you a mark, but at A Level, you need to think about it. Explain it, Describe it and give an Example **(EDE)**

Purpose of an Input Device

Purpose of an Output Device

Purpose of an Storage Device

Use the space below to detail as many input, output and storage devices as you can.

Input Devices	Output Devices	Storage Devices

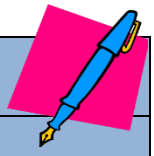
Heathcote School and Science College

Systems Software and Applications Software

When we talk about software there are also many different types, here we need an explanation of the different types of software and the DIFFERENCES between the two.

Systems Software

Applications Software



What are the differences between system software and application software	
Systems Software	**Application Software**

Heathcote School and Science College

For the exam

Topic	Understand	Could Improve	Needs Work
Number Systems			
express numbers in binary, binary-coded decimal (BCD), octal and hexadecimal;			
Data Capture, Preparation and entry			
describe and use two's complement and sign and magnitude to represent negative integers;			
perform integer binary arithmetic, that is addition and subtraction;			
explain the use of code to represent a character set (ASCII, EBCDIC and UNICODE);			
describe manual and automatic methods of gathering and inputting data into a system, including form design, keyboard entry, voice recognition, barcodes, optical mark recognition (OMR), optical character recognition (OCR), magnetic ink character recognition (MICR), touch screens; image capture, chip and pin, sensors and remote data logging			
Validation and verification of data			
explain the techniques of validation and verification, and describe validation tests which can be carried out on data;			
Outputs from a system			
Describe possible forms of output such as graphs, reports, interactive presentations, sound, video, images, animations, stating the advantages and disadvantages of each with reference to the target audience;			
Explain the procedures involved in backing up data and archiving, including the difference between data that is backed up and data that is archived.			

Heathcote School and Science College

Number Systems

Defining the terms

When talking about size of files it is important to understand how the sizes go up. So this next section is all about defining how the sizes go up and how they relate to each other.

Take each of the key sizes, put them in order, then tell me how they relate.

Terabyte, kilobyte, bit, byte, nibble, gigabyte, megabyte,

Size	How it relates to it's predecessor

Extension Task: What comes after a terabyte? Do a bit of research, see if you can find out if you were to put all the information in the world together, how much data would that be? I want a web link and some thoughts about his.

Why do the sizes jump up in 1024 and not 1000?
What is Binary?
In the space below, write out the binary system up to 15 Places

Heathcote School and Science College

Number Conversion

This section uses worksheets, so this workbook will only help in your revision by writing out the steps you have to take in order to perform a conversion.

There are 10 types of people in this world

(those who are sick of this joke and those that still don't get it!) hopefully, by the end of this section you should be in the former, not the latter!

Write out the steps in order to convert 00101101 to Denary	
1	
2	
3	

Write out the steps in order to convert 67 to Binary	
1	
2	
3	

Task: Explain the joke in the heading to this section

Binary Coded Decimal – Making something simple harder

So, you get binary, well BCD does something slightly different, but still uses a binary notation

Define: Binary Coded Decimal

What are the positives and negatives of using Binary Coded Decimal

Positives of BCD	Negatives of BCD

Octal – Are ya feeling lucky?

So imagine now that we have 2 number systems, Denary (0-9) and binary (0-1), but then you think about it, how many bits are there in a byte? Then we get to octal!

Define: Octal

Write out the steps in order to convert a binary number into Octal (e.g. **10110011**)	
1	
2	
3	
4	

So far, it's pretty easy, but now it gets the brain working. What if I want to convert decimal to octal? (you could break the decimal to binary, then back up to octal, but that's too easy!)

Write out the steps to convert 891(D) to Octal	
1)	
2)	
3)	
4)	

I put a HEX on you

Just when you thought it couldn't get any more confusing having binary and denary number systems, you now have to talk about ANOTHER one. Hexadecimal

Define: Hexadecimal

Write out the steps in order to convert a binary number into Hexadecimal (e.g. **10110011**)	
1	
2	
3	
4	

What would you do if you needed to convert a Hexadecimal number **01A** into Decimal	
1	
2	
3	
4	

Write out the steps to convert a decimal number to Hex	
1)	
2)	
3)	
4)	

Are there any underlying rules that you can remember about number base conversion?

Negative numbers

When talking about negative numbers there are 2 different schools or thought. If you think about computers all of them break down to electronics. So the way that the electronics represent negative numbers can make a big difference.

Sign and Magnitude

How does Sign and Magnitude represent negative numbers?
What is the issue with this?
Show how you convert a negative number into this system (e.g. -75)

2's Compliment – Nice Hat!

So we've realised that there are 2 ways of representing binary numbers, but neither is perfect, so 2's compliment was invented, the perfect compliment to a 2 base system.

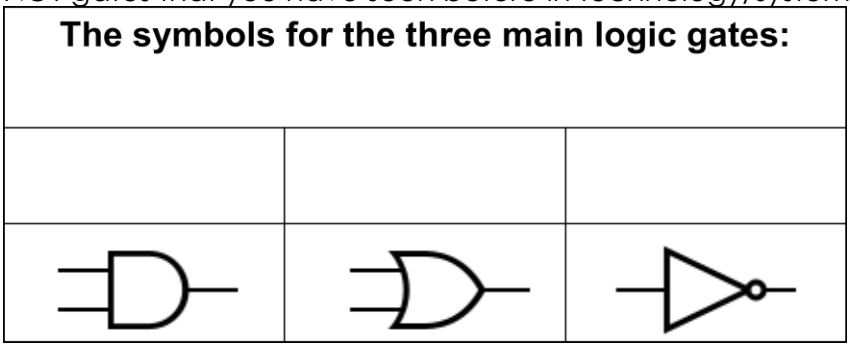
How does 2's compliment work?
What is the benefit of this?
How do you perform a 2's compliment notation on a number? (e.g. -75)

Heathcote School and Science College

Binary Arithmetic (and a bit of electronics!)

All computers at the base level have to work on electronics, this goes back to the AND, OR and NOT gates that you have seen before in technology/systems.

The symbols for the three main logic gates:



What is meant by a truth table?

AND

Input		Output

OR

Input		Output

NOT

Input		Output

Binary Arithmetic

Describe the process of binary addition
Show with an example how to add 25 to 78

Describe the process of binary subtraction
Show with the example how to subtract 35 from 67

Heathcote School and Science College

Character Sets

ASCII

What is ASCII?
What is it used for?
How many bits does an ASCII letter use? Show an example.

Using the ASCII character set at the back of this book, I want you to translate the following words into Binary.

Word	Binary
GCSE	
Hello	
World	
Computing	
Your name	

What's really key, is if you can write your own ASCII codes, then you have a secret encryption standard that only you can translate (well, you and anyone who knows ASCII!)

Unicode

What is UNICode?
Why was this made?
What is the point in having 2 character sets?

Find a RELIABLE and USEFUL website that contains the Unicode character set and place it in the space below for reference when revising.

Extension: The maximum number that ASCII can represent is 255, why is this significant?

EBCDIC

What is EBCDIC?

What is it used for? What is it's history?

What are the advantages of EBCDIC?

What are the disadvantages?

Find an EBCDIC character table and place the link below:

Heathcote School and Science College

Image Representation

Computers are weird, we already know that all they do is calculations and that all letters are really just numbers in disguise, so what happens when we want to talk about images? You use images on a computer all the time, from your digital camera to the wallpaper on your desktop, so how does a computer represent this?

Images can be classified into 2 main areas, Vectors and Bitmaps. Detail the differences below	
Vector	**Bitmap**

There are loads of different types of image, detail them below AND state what they are best used for:

No.	Image Type	Description/Best used for

What is Meta-data? How does this fit in with image representation?

Extension task: With phones now taking a vast amount of our pictures for us, what extra meta-data can they store? For additional marks you could discuss weather this is a good or bad thing?

Heathcote School and Science College

The matrix has you

This may sound like a bit of a joke title, but it's a good way to learn how an image is broken down by the computer. Detail how this works below.

How does a computer break down an image? How is that image then represented on the computer screen? (in simple terms talk about a black and white image)

What about a colour image, how does this make it more complicated?

The matrix – resolutions

Resolution is really important when you are talking about the quality of an image.

What is resolution? Why is resolution important?

What is the downside of a high resolution

Extension Task: What is the maximum number of pixels the human eye can see before it is useless? (This would be useful to know when buying your next digital camera!)

Extension Task: What is a Mega-Pixel? Why is a mega-pixel important when you are buying a digital camera/camera phone?

Extension Task: What other things are important when buying a digital camera/camera phone?

Heathcote School and Science College

Sound Representation

The way our ears work are different to the way a computer hears. If you think about GCSE Biology/physics, you know that we hear in a wave form (A wibbly wobbly wavey thing). Computers can't do it like that….so we have to think of a way of translating the work from analogue to digital (and vice-versa), think digital radio!

Keywords	Description
Digital Signals	
Analogue Signals	
Sample Size	
Bit rate	

How does an Analogue signal work?

How does the computer convert analogue to digital?

What Factors affect the quality of the sound?(and why)

Heathcote School and Science College

Using the code below, Decode the songs given to you in class

1	2	3	4	5	6	7	8	9	10	11	12	13
a	b	c	d	e	f	g	h	i	j	k	l	m
14	15	16	17	18	19	20	21	22	23	24	25	26
n	o	p	q	r	s	t	u	v	w	x	y	z

Letter codes (English)

Song 1	
Song 2	
Song 3	
Song 4	
Song 5	
Song 6	

Explain in the space below how you decoded the songs.

Heathcote School and Science College

Data Compression

Throughout this unit we have talked about how things are represented in a computer. Now, if you look at any of your files, be it sound or images, you can see the different types. What really makes it interesting is that all the files are different sizes, so the same image saved as a JPG and a GIF will be different sizes. How can this be? The answer….compression

What is compression?

What are the different types of Compression?		
No	Type	Where/How Used
2		
1		

Extension: Can you explain how Lossy compression works with Sound or Images?

What are the key difference between lossy and lossless	
Lossy	**Lossless Compression**

Describe: Lossy Compression

Describe: Lossless Compression

Data Compression Exercise 1

The following example gives an alternative representation of **Compressed Text**; your task is to decompress it.

- Copy text systematically from the *Compressed Text* box to the *Expanded Text* box below. The first few words have already been copied.
- Each pair of numbers in brackets represents a piece of text that has already appeared, and you should replace it with the original text.
- The numbers indicate where to find that text:
  - **First number**: The position of the first character to be copied
  - **Second number**: The position of the last character to be copied
- For example when you come across (9,15) you copy down the character at position 9 in the ***Expanded Text*** (a space) followed by the next 6 characters (w, o, o, etc) up to the character at position 15 (c).

Compressed Text

```
How much wood could a(9,15)hopper(27,31) if(20,36)(16,18)(26,31)(9,13)
```

Expanded Text

```
01          06          11          16          21          26          31          36
How much wood
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

   41          46          51          56          61          66          71
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
```