Computer Science A'level
Hand Book

All you need to know . . .

Contents

Welcome to Computer Science!

Assessment overview

Students must take all three components to be awarded the OCR A Level in Computer Science.

Component	Marks	Duration	Weighting	
Computer systems (01)	140	2 hours 30 mins	40%	The internal workings of the (CPU), data exchange, software development, data types and legal and ethical issues. Calculators not allowed.
Algorithms and programming (02)*	140	2 hours 30 mins	40%	Using computational thinking to solve problems. Calculators not allowed.
Programming project (03)*	70	-	20%	Non-exam assessment. Students will be expected to analyse a problem (10 marks), and design (15 marks), develop and test (25 marks), and evaluate and document (20 marks) a program. The program must be to solve it written in a suitable programming language.

* Indicates the inclusion of synoptic assessment.

Content overview

Component 01: Computer systems

Students are introduced to the internal workings of the (CPU), data exchange, software development, data types and legal and ethical issues. The resulting knowledge and understanding will underpin their work in component 03.

It covers:

- The characteristics of contemporary processors, input, output and storage devices
- Types of software and the different methodologies used to develop software
- Data exchange between different systems
- Data types, data structures and algorithms
- Legal, moral, cultural and ethical issues.

Component 02: Algorithms and programming

This builds on component 01 to include computational thinking and problem-solving.

It covers:

- What is meant by computational thinking (thinking abstractly, thinking ahead, thinking procedurally etc.)
- Problem solving and programming – how computers and programs can be used to solve problems
- Algorithms and how they can be used to describe and solve problems.

Component 03: Programming project

Students are expected to apply the principles of computational thinking to a practical coding programming project. They will analyse, design, develop, test, evaluate and document a program written in a suitable programming language. The project is designed to be independently chosen by the student and provides them with the flexibility to investigate projects within the diverse field of computer science. We support a wide and diverse range of languages.

Personal Learning Check List

Code	AS Level	A Level	Specification point description	Complete
SLR2	1.1.2a	1.1.2a	The differences between, and uses of, CISC and RISC processors	
SLR2	1.1.2b	1.1.2c	Multicore and parallel systems	
SLR3	1.1.3a	1.1.3a	How different input output and storage devices can be applied as a solution of different problems	
SLR3	1.1.3b	1.1.3b	The uses of magnetic, flash and optical storage devices	
SLR3	1.1.3c	1.1.3c	RAM and ROM	
SLR3	1.1.3d	1.1.3d	Virtual storage	
SLR4	1.2.1a	1.2.1a	The need for, function and purposes of operating systems	
SLR4	1.2.1b	1.2.1b	Memory management (paging, segmentation and virtual memory)	
SLR4	1.2.1c	1.2.1c	Interrupts, the role of interrupts and Interrupt Service Routines (ISR), role within the fetch decode execute cycle	
SLR4	1.2.1d	1.2.1d	Scheduling: round robin, first come first served, multi-level feedback queues, shortest job first and shortest remaining time	
SLR4	1.2.1e	1.2.1e	Distributed, embedded, multi-tasking, multi-user and real time operating systems	
SLR4	1.2.1f	1.2.1f	BIOS	
SLR4	1.2.1g	1.2.1g	Device drivers	
SLR4	1.2.1h	1.2.1h	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	
SLR5	1.2.2a	1.2.2a	The nature of applications, justifying suitable applications for a specific purpose	
SLR5	1.2.2b	1.2.2b	Utilities	
SLR5	1.2.2c	1.2.2c	Open source vs Closed source	
SLR5	1.2.2d	1.2.2d	Translators: interpreters, compilers and assemblers	

SLR8	1.2.3a	1.2.3a	Procedural programming language techniques: program flow, variables and constants, procedures and functions, arithmetic, Boolean and assignment operators, string handling, file handling	
SLR8	1.2.3b	1.2.3b	Assembly language (including following and writing programs with Little Man Computer)	
SLR10	1.3.1a	1.3.2a	Relational database, flat file, primary key, foreign key, secondary key, entity relationship modelling, normalisation and indexing	
SLR10	1.3.1b	1.3.2b	Methods of capturing, selecting, managing and exchanging data	
SLR11	1.3.2a	1.3.3a	Characteristics of networks and the importance of protocols and standards	
SLR11	1.3.2b	1.3.3b	Internet structure: -The TCP/IP Stack -DNS -Protocol layering -LANs and WANs -Packet and circuit switching	
SLR11	1.3.2c	1.3.3e	Client-server and Peer to Peer	
SLR12	1.3.3a	1.3.4a	HTML, CSS and JavaScript	
SLR12	1.3.3b	1.3.1a	Lossy vs lossless compression	
SLR13	1.4.1a	1.4.1a	Primitive data types, integer, real/floating point, character, string and Boolean	
SLR13	1.4.1b	1.4.1b	Represent positive integers in binary	
SLR13	1.4.1c	1.4.1c	Use of sign and magnitude and two's complement to represent negative numbers in binary	
SLR13	1.4.1d	1.4.1d	Addition and subtraction of binary integers	
SLR13	1.4.1e	1.4.1e	Represent positive integers in hexadecimal	
SLR13	1.4.1f	1.4.1f	Convert positive integers between binary hexadecimal and denary	
SLR13	1.4.1g		Positive and negative real numbers using normalised floating point representation	
SLR13	1.4.1h	1.4.1j	How character sets (ASCII and UNICODE) are used to represent text	
SLR14	1.4.2a	1.4.2a	Arrays (of up to 3 dimensions), records, lists, tuples	

SLR14	1.4.2b		The properties of stacks and queues	
SLR15	1.4.3a	1.4.3a	Define problems using Boolean logic	
SLR15	1.4.3b	1.4.3b	Manipulate Boolean expressions, including the use of Karnaugh maps to simplify Boolean expressions	
SLR15	1.4.3c	1.4.3d	Using logic gate diagrams and truth tables	
SLR16	1.5.1a	1.5.1a	The Data Protection Act 1998	
SLR16	1.5.1b	1.5.1b	The Computer Misuse Act 1990	
SLR16	1.5.1c	1.5.1c	The Copyright Design and Patents Act 1988	
SLR16	1.5.1d	1.5.1d	The Regulation of Investigatory Powers Act 2000	
SLR18	2.1.1a	2.1.1a	The nature of abstraction	
SLR18	2.1.1b	2.1.1b	The need for abstraction	
SLR18	2.1.1c	2.1.1c	The differences between an abstraction and reality	
SLR18	2.1.1d	2.1.1d	Devise an abstract model for a variety of situations	
SLR19	2.1.2a	2.1.2a	Identify the inputs and outputs for a given situation	
SLR19	2.1.2b	2.1.2b	Determine the preconditions for devising a solution to a problem	
SLR19	2.1.2c	2.1.2d	The need for reusable program components	
SLR20	2.1.3a	2.1.3a	Identify the components of a problem	
SLR20	2.1.3b	2.1.3b	Identify the components of a solution to a problem	
SLR20	2.1.3c	2.1.3c	Determine the order of the steps needed to solve a problem	
SLR20	2.1.3d	2.1.3d	Identify sub-procedures necessary to solve a problem	
SLR21	2.1.4a	2.1.4a	Identify the points in a solution where a decision has to be taken	
SLR21	2.1.4b	2.1.4b	Determine the logical conditions that affect the outcome of a decision	
SLR21	2.1.4c	2.1.4c	Determine how decisions affect flow through a program	

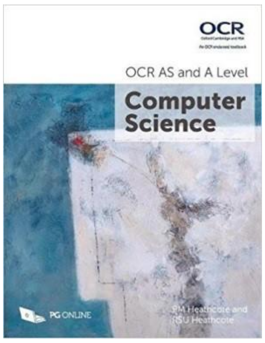
Lesson Expectations

All students to be issued with exercise books to take notes; different exercise books for each teacher. Exercise books must be used in line with school policy. This includes titles and long dates each lesson. Titles should include referential links to spec points. E.g. 1.1.1a-1.1.1d. Students should be encouraged to highlight keywords where appropriate. Students should print off work that is done on the computer and stick in.

Resources needed:

Students should purchase the following text book:

OCR AS and A Level Computer Science (**ISBN-13:** 9781910523056)



An electronic copy is the staff shared area for staff and if students prefer to use an electronic copy then that is fine as well.

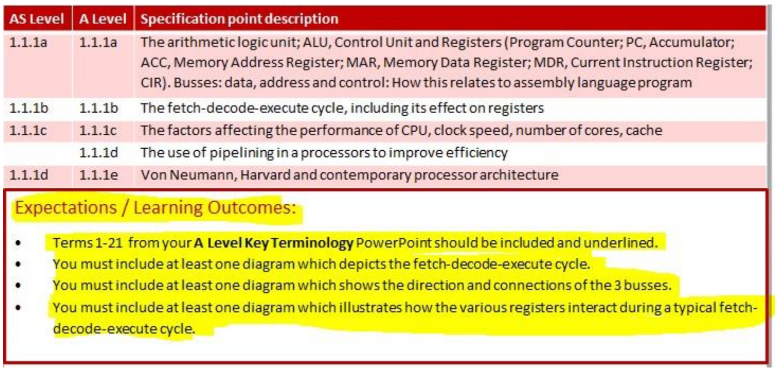
Green pens for peer/self assessment.

Homework

Students to be set meaningful homework that consolidates learning from each lesson. Remember one hour of homework for each hour of lesson time. This can take the form of:

Definitions from the terminology document for content covered. Definitions should be written into the back of the exercise book like a glossary.

Students to create a double spread (A3) of the SLR notes in their exercise book. This should cover the expectations/ learning outcomes from the SLR document (highlighted area to the right). SLR marking documents to be stuck into books.



AS Level	A Level	Specification point description
1.1.1a	1.1.1a	The arithmetic logic unit; ALU, Control Unit and Registers (Program Counter; PC, Accumulator; ACC, Memory Address Register; MAR, Memory Data Register; MDR, Current Instruction Register; CIR). Busses: data, address and control: How this relates to assembly language program
1.1.1b	1.1.1b	The fetch-decode-execute cycle, including its effect on registers
1.1.1c	1.1.1c	The factors affecting the performance of CPU, clock speed, number of cores, cache
	1.1.1d	The use of pipelining in a processors to improve efficiency
1.1.1d	1.1.1e	Von Neumann, Harvard and contemporary processor architecture

Expectations / Learning Outcomes:

- Terms 1-21 from your **A Level Key Terminology** PowerPoint should be included and underlined.
- You must include at least one diagram which depicts the fetch-decode-execute cycle.
- You must include at least one diagram which shows the direction and connections of the 3 busses.
- You must include at least one diagram which illustrates how the various registers interact during a typical fetch-decode-execute cycle.

Coding

Where programming exercises have been done, they need to be printed in colour and stuck in. If comments are missing, students need to hand annotate the different programming techniques used. Programming activities should be linked to unit 2 SLR activities.

Extend your learning

Places to visit

Looking for places to visit to expand your understand of Computer Science try:

Bletchley Park - https://bletchleypark.org.uk/

Science Museum - https://www.sciencemuseum.org.uk/

Websites to watch

https://www.ocr.org.uk/qualifications/as-and-a-level/computer-science-h046-h446-from-2015/

https://www.memrise.com/course/1211053/ocr-a-level-computer-science-2/

https://student.craigndave.org/a-level-videos

http://teach-ict.com/

https://teachyourselfcs.com/

Computer Science Task

To ensure that you are fully ready for your first lessons in Computer Science you need to complete the following:

Task 1:

Complete 'Learn the Basics' on https://www.learnjavaonline.org/

- Hello, World!
- Variables and Types
- Conditionals
- Arrays
- Loops
- Functions
- Objects
- Compiling and Running with Arguments

You should make notes and take screen captures of your code and add to this booklet.

Task 2:

A thief has managed to find out the four digits for an online PIN code but doesn't know the correct sequence needed to hack into the account.

Design and write a program that displays all the possible combination for any four numerical digits entered by the user. The program should avoid displaying the same combination more than once.

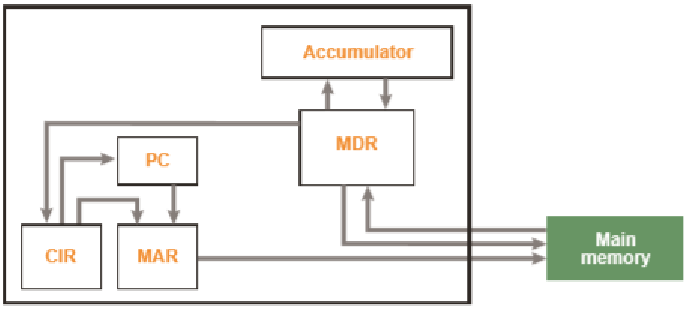
Task 3:

You have a budget of only £950. You are to list the hardware you are to select to build this gaming PC.

- What is clock speed?
- What is cache?
- What happens within a CPU Cycle?
- What advantages are there of setting your clock speed to be faster (overclocking it)?
- What are the dangers of overclocking a system?
- What is a GPU and how does it differ from a CPU?
- What are the differences between magnetic, solid state and optical storage?

Task 4:

Explain how the internals of a CPU work. What are the differences and applications of the two CPU architectures Von Neumann and Harvard architectures?



Task 5:

Explain how the internals of a CPU work. What are the differences and applications of the two CPU architectures Von Neumann and Harvard architectures?

Task 6a:

To create a web page you need to mark it up in HTML. Use the tutorials provided by w3schools, http://www.w3schools.com/html, to help you to create a simple web page in HTML that defines a form for a user to enter sign in details for a customer of an online business. You need to consider the data required, but this must contain some form of ID or username and password.

Task 6b:

Adding a suitable style sheet will add consistency to any pages produced. Referring to the resource by w3schools: http://www.w3schools.com/html/html_css.asp.

Add suitable style sheet commands to format the web page you have created.

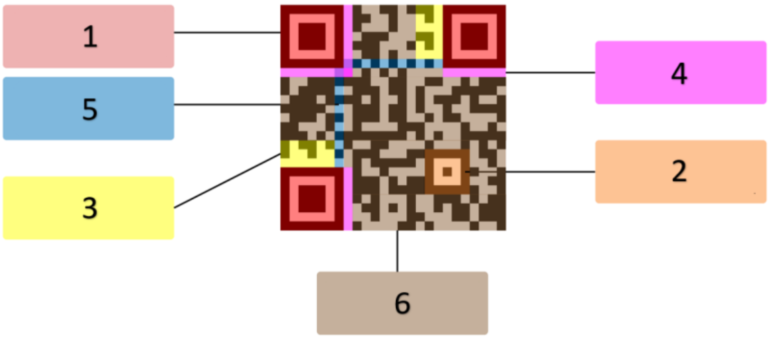
Task 6c:

To add interactivity to the webpages use javascript. Use the w3 resources, http://www.w3schools.com/html/html_scripts.asp, to help you with this task.

Add validation to ensure customer IDs are of the right length and type and that all fields have been completed.

Task 7 – optional extension

Research how QR Codes work.



Task 8 – optional extension

This task will help you towards A-Level Component 3 Programming Project. Research how to install Python and install the graphics library called PyGame. Research how to code in Object Oriented Programming (OOP) and create a retro game of your choosing.

https://www.python.org

https://www.pygame.org

http://programarcadegames.com

Task 9 – optional extension

Complete the extension page on a future technology.

Please email Mr Stewart if you have a query regarding the activities
mstewart@rodingvalley.net

Good luck and enjoy!

Task 1:

Hello World Evidence:

Variables and Types Evidence:

Conditions Evidence:

Arrays Evidence:

Loops Evidence:

Functions Evidence:

Arguments Evidence:

Compiling and Running with Arguments Evidence:

Task 2 Evidence:

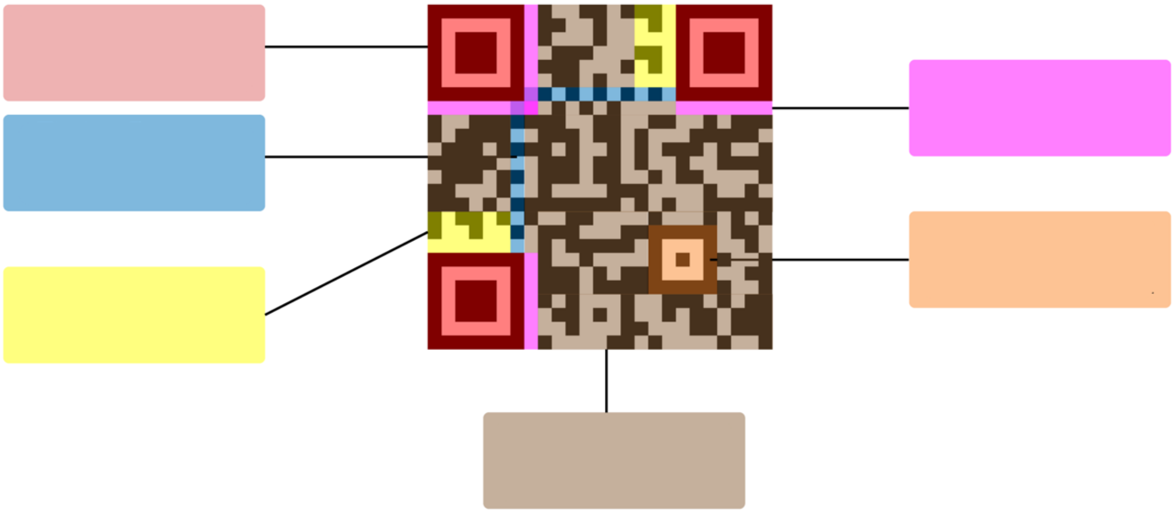
Task 3 Evidence:

Task 4 Evidence:

Task 5 Evidence:

Task 6 Evidence:

Task 7 – optional extension



Task 8 – optional extension

Computer Science and Ethics - extension

Read the BBC New article 'Biased and wrong, facial recognition tech in the dock - https://www.bbc.co.uk/news/business-48842750 or use the QR code and then Outline your views on facial recognition below

What is facial recognition technology and what is it being used for (according the article)

If you hover your phone over this QR code, your smart phone will link you to the article you need!

Outline positive uses of FR

Outline negative issues of the use of FR