



**WELCOME
TO A LEVEL COMPUTER
SCIENCE**

TODAY

Welcome

Specification

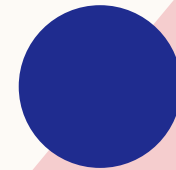
Assessment

Day to day lessons

Transition Task

Resources

Programming fun





Mrs Murfin
(Head of the computer
Science department)



Mrs Jacobs
(Computer Science
teacher)

WELCOME TO COMPUTER SCIENCE

The Computer Science course is 10 lessons over a 2 week cycle (teaching time) taught in room 201.

The lessons are split between Mrs Murfin and Mrs Jacobs (5 lessons each over 2 weeks).

Mrs Murfin teaches paper 2 and a section of paper 1 (exchanging data) & the project.

Mrs Jacobs teaches paper 1 & the project

SPECIFICATION

Computer systems (01)	Algorithms and programming (02)	Programming project 03
1.1 The characteristics of contemporary processors	2.1 Elements of computational thinking	Analysis of the problem
1.2 Software and software development	2.2 Problem solving and programming	Design of the solution
1.3 Exchanging data	2.3 Algorithms to solve problems and standard algorithms	Developing the solution
1.4 Data types, data structures and algorithms		Evaluation
1.5 Legal, moral, cultural and ethical issues		

Download the specification for more information
<https://www.ocr.org.uk/Images/170844-specification-accredited-a-level-gce-computer-science-h446.pdf>



1.1 The characteristics of contemporary processors, input, output and storage devices	
Components of a computer and their uses	
1.1.1 Structure and function of the processor	(a) The Arithmetic and Logic Unit; ALU, Control Unit and Registers (Program 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. (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. (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.

ASSESSMENT

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 <i>The learner will choose a computing problem to work through according to the guidance in the specification.</i> - 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* – Moderated upload or 04* – Moderated postal or 80 – Carry forward (2018 onwards)* 70 marks Non-exam assessment	20% of total A level

DAY TO DAY LESSONS

In computer Science we use a flip learning approach to deliver the course

How Flipped Learning Work?

1. Before Class:

You will watch the Craig and Dave video lectures This introduces you to new concepts at your own pace.

You will be expected to make notes using the cornel notes method or use your own method of note taking on the set video.

2. During Class: (following lessons)

Class time is used for discussions, problem-solving, group work, and hands-on activities. At this point we will provide guidance and personalized support.

3. After Class:

You may reflect, complete projects, or extend your learning by reading around topics.

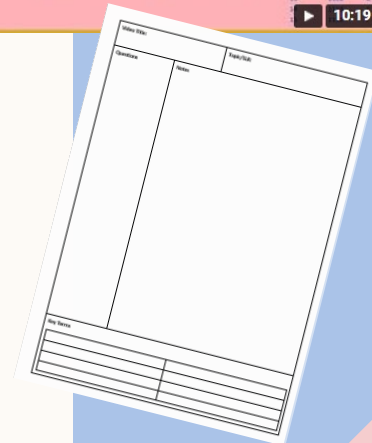
Benefits of flipped learning

Encourages active learning and student engagement

Allows for personalized instruction

Frees up class time for collaboration and deeper understanding

Students can review materials as needed



ASSESSMENTS: IN CLASS

SLRS (student learning records)

At the end of every topic you will complete a student learning record that is mapped back to the specification. These SLRs are graded by your teachers.

Here is an example of a SLR

SLR1 (AS & A) - Structure and function of the processor

End of unit assessments

At the end of each unit you will be assessed by your teacher

Mock exams

You will have at least 1 mock exam in the spring/summer term based on everything you have learnt the year so far,

The image shows two overlapping student learning record (SLR) forms. The top form is titled 'AS/A Unit 1 | SLR 1 | Structure and function of the processor' and contains a table of specification points and learning objectives. The bottom form is also titled 'AS/A Unit 1 | SLR 1 | Structure and function of the processor' and contains a table of minimum expectations and learning outcomes, a feedback section, and a comment and action required section.

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). Buses: 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 processor to improve efficiency.
1.1.1d	1.1.1e	Von Neumann, Harvard and contemporary processor architecture.

Resources

PG Online textbook page ref: 1-12
Hodder textbook page ref: 124-133
[Craig 'W' Dave videos for SLR 1](#)

Minimum expectations and learning outcomes

Target: ☐ Overall grade: ☐

Feedback

Breadth	Depth	Presentation	Understanding
☐ All	☐ Analyzed	☐ Excellent	☐ Excellent
☐ Most	☐ Explained	☐ Good	☐ Good
☐ Some	☐ Described	☐ Fair	☐ Fair
☐ Few	☐ Identified	☐ Poor	☐ Poor

Comment and action required

The image shows a mock exam paper for Computer Science, Paper 1: Principles of Computer Science. The paper is dated Monday 13 May 2019 and is for Level 1/Level 2 (GCSE/9-1). It includes a section for candidate details, a section for instructions, and a section for questions.

Instructions

- Use black ink or ball-point pen.
- Write in the boxes at the top of this page with your name, centre number and candidate number.
- Answer all questions.
- Answer the questions in the spaces provided - show any rough working in the margin.
- You are not allowed to use a calculator.

Information

- The total mark for this paper is 80.
- The marks for each question are shown in brackets - use this as a guide to how much time you should spend on each question.

Advice

- Read each question carefully before you start to answer it.
- Try to answer every question.
- Check your answers: There is time at the end.

Benefits of flipped learning

Encourages active learning and student engagement

Allows for personalized instruction

Frees up class time for collaboration and deeper understanding

Students can review materials as needed

TRANSITION TASK

Access the transition task via this link.

The transition task will give you a taster of a theory lesson in computer Science.

<https://www.smrt.bristol.sch.uk/hiddenarea/transition-tasks-for-new-students-draft-2025>

Data Representation Task 1: Watch

To begin with I would like you to watch a series of videos :

[Video 1 : Units](#)

[Video 2: Converting between denary & binary](#)

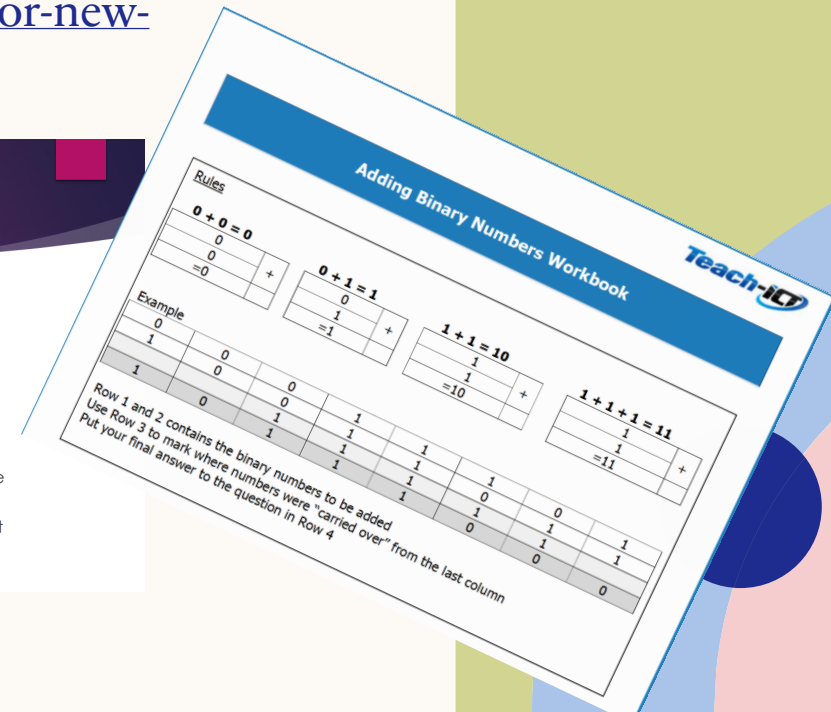
[Video 3: Adding two 8 bit binary numbers](#)

[Video 4: Binary shifts](#)

This will introduce you to the Denary and Binary number systems and how we can convert between them and add them together.

It is important to take in the information in the videos. Please watch them first and just concentrate on what is being said.

Computer Science
TRANSITION WORK



RESOURCES

Learn python

This is an Introduction to Programming course and the Advanced Course in Programming from the Department of Computer Science at the University of Helsinki.

<https://programming-24.mooc.fi/part-1/1-getting-started>

Another course you could use to help you learn python

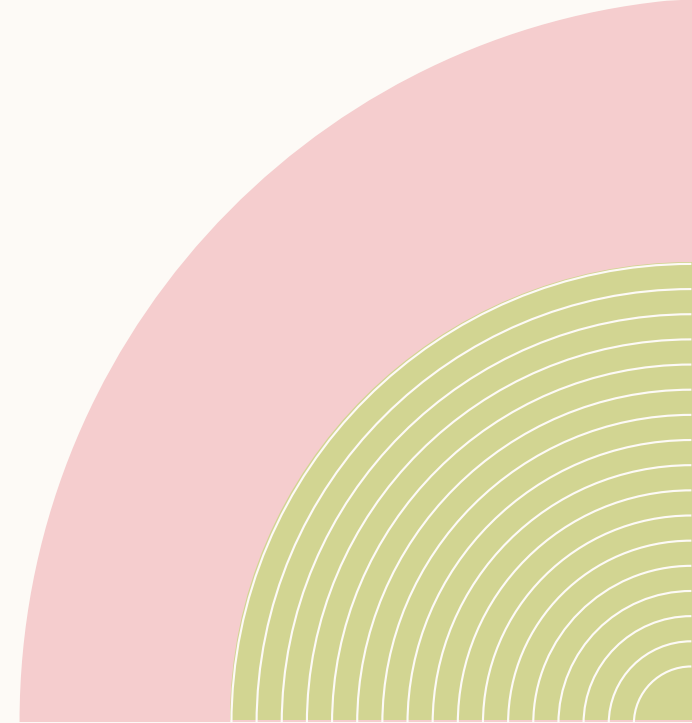
<https://www.w3schools.com/python/>

Craig and Dave website and resources

<https://craigndave.org/>

Course textbook

- OCR AS and A Level Computer Science by Pm Heathcote
- Essential algorithms and data structures by Mr David Hillyard (Author), Mr Craig Sargent (Author), Mr Andrew Fenn (Editor)

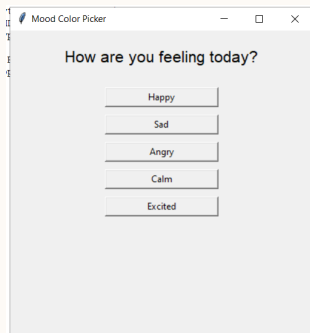


PROGRAMMING FUN

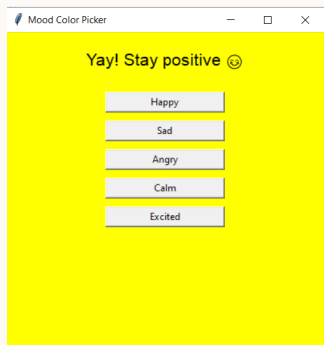
MOOD APP

Advanced programmers

Create a mood app in python using the GUI Tkinter



This is an example of the first screen you need to code.



When you click on a button the screen will change colour and a message will appear above the buttons

Intermediate Programmers

Open the file below in python and complete the following

- Change the colours
- Change the buttons name
- Change the messages above the button

Here is the file

Mood app

PLEASE CONTACT US IF YOU HAVE ANY QUESTIONS

Please contact us if you have any questions

murfinr@smrt.bristol.sch.uk

(head of Computer Science department)

jacobss@smrt.Bristol.sch.uk

(Computer Science Teacher)