Computer Science Unit Overview Year 11 Term 1.1

The exam boards have suggested that the non-examined Assessment programming work which is normally required to be completed in school over 20 hours be completed outside of the school setting, which will free up additional time in class to revise and consolidate work previously completed remotely. This will enable the remainder of the full specification to be covered during year 11. The scheme of work has been designed to identify and support gaps and misconceptions by re-visiting skills and building on previous knowledge frequently during the course.

Low stakes testing will continue to take place using BBC Bitesize topic tests with the results recorded in Class Notebook, so that gaps in knowledge can be identified and individual support in the form of one to one/small group catch up sessions will be given to students struggling to cover the course content.

Key Knowledge which we will re-visit during year 11 include

11..1 Data representation - text

11.2 Ethical impact

12.1 Data representation

12.2 Reading and writing to files

1.2 Dictionaries

2.2 Programming

3.2 Bitmaps

6.1 Hardware - CPU

7.1 Hardware - Memory

7.1.1 Hardware - More on Memory

7.2 CSV files

8.2 Writing to CSV files

Computers are able to store and manipulate large quantities of data. They use binary to represent different types of data. Students are expected to learn how different types of data are represented in a computer.

Students must be familiar with the hardware and software components that make up a computer system and recognise that computers take many forms from embedded microprocessors to distributed clouds.

Computer Science - Year 11				
What are we learning?	**What knowledge, understanding and skills will we gain?**	**What does mastery look like?**	**How does this build on prior learning?**	**What additional resources are available?**
Data: Data storage and Compression Computers: Hardware, Data storage	**Knowledge:** bit, nibble, byte, kilobyte (KB), megabyte (MB), gigabyte (GB), terabyte (TB) What is data compression Methods of compressing data (lossless, lossy) JPEG and MP3 are examples of lossy algorithms What is a lossless, run-length encoding (RLE) algorithm Physical data storage devices (magnetic, optical, solid state) What is cloud storage, contemporary secondary storage **Understanding:** Understand how to convert between the terms 'bit, nibble, byte, kilobyte (KB), megabyte (MB), gigabyte (GB), terabyte (TB)' Understand the need for data compression and methods of compressing data Understand how a lossless, run-length encoding (RLE) algorithm works Understand how data is stored on physical devices (magnetic, optical, solid state)	Be able to convert between the terms 'bit, nibble, byte, kilobyte (KB), megabyte (MB), gigabyte (GB), terabyte (TB)' Demonstrate how to convert between 'bit, nibble, byte, kilobyte (KB), megabyte (MB), gigabyte (GB), terabyte (TB)' Explain the need for data compression and methods of compressing data Describe how a lossless, run-length encoding (RLE) algorithm works Describe how data is stored on physical devices (magnetic, optical, solid state) Explain the concept of storing data in the 'cloud' and other contemporary secondary storage	8.2 Binary Bits and Bobs Y10 Summer Term lesson 5.1, 6.1, 7.1, 8.1	cs4fn, Queen Mary, University of London (www.cs4fn.org). Computer Science Unplugged (http://csunplugged.org/ Magnetic storage, e.g. a hard disk drive: http://en.wikipedia.org/wiki/File:Laptop-hard-drive-exposed.jpg Optical storage, e.g. a CD or DVD drive: http://en.wikipedia.org/wiki/File:Dvdburner.jpg Solid state, e.g. a solid state drive: http://en.wikipedia.org/wiki/File:Vertex_2_Solid_State_Drive_by_OCZ-top_oblique_PNr%C2%B00307.jpg How data is stored magnetically: http://richannel.org/christmas-lectures/2008/2008-chris-bishop#/christmas-lectures-2008-chris-bishop--the-ghost-in-the-machine (12.00–13.45 minutes) Labelled diagram of a hard disk drive: http://en.wikipedia.org/wiki/File:Hard_drive-en.svg Image of the surface of a hard disk drive: http://en.wikipedia.org/wiki/File:Aufnahme_einzelner_Magnetisierungen_gespeicherter_Bits_auf_einem_Festplatten-Platter..jpg. Reading head movement on a CD (http://en.wikipedia.org/wiki/File:Cd-rom-drive-reading-head-movement.gif) The magnified pits on a CD (http://en.wikipedia.org/wiki/File:Compact_disk_data_layer_2d_3d.PNG) 'Explaining Solid State Disks' video: http://www.youtube.com/watch?v=viac3j6MeII (0.00 – 2.40 minutes) 'Where is Google?' Map of datacentre locations (https://www.google.com/about/datacenters/inside/locations/index.html) General resources *BCS Glossary of Computing and ICT*, 13th edition (ISBN 9781780171500)

	Understand the concept of storing data in the 'cloud' and other contemporary secondary storage **Skills:** Convert between the terms 'bit, nibble, byte, kilobyte (KB), megabyte (MB), gigabyte (GB), terabyte (TB)'			A range of articles on teaching coding: www.edsurge.com/guide/teaching-kids-to-code Python resources Official Python documentation (also available through help in IDLE): www.docs.python.org/3/ Python summer school from Anglia Ruskin University is an excellent resource with videos and programming challenges: http://www.pythonschool.net/ Python code for kids is a clearly written summary of the Python language written in accessible language: www.pythondictionary.code-it.co.uk/ 'Python in 10 minutes' is a quick run through of the basic concepts: www.korokithakis.net/tutorials/python/ 'Quintin Cutts – Too much doing, not enough understanding' is a 20-minute video containing useful ideas and concepts on how to teach programming: www.youtube.com/watch?v=Pim4aYfiZiY Free online books *Think Python How to Think Like a Computer Scientist* is a free online book with programming challenges at the end of each chapter: www.greenteapress.com/thinkpython/thinkpython.pdf *A Byte of Python* is an excellent online book, though it does not use IDLE as the editor: www.swaroopch.com/notes/Python/ Books *Python Programming for the Absolute Beginner*, M. Dawson (published Course Technology 2010) (ISBN 9781435455009) is an excellent book with clear explanations of each bit of code together with free downloads and example games built into each chapter. *Python for Kids: A Playful Introduction to Programming*, J. Briggs (published No Starch Press 2013) (ISBN 9781593274078).

Computer Science Unit Overview Year 11 Term 1.2

Students must be familiar with the hardware and software components that make up a computer system and recognise that computers take many forms from embedded microprocessors to distributed clouds.

Computer networks and the internet are now ubiquitous. Many computer applications in use today would not be possible without networks. Students should understand the key principles behind the organisation and of computer networks. Ideally, they should be able to experiment by setting up a simple network.

Students should be aware of the influence of computing technology and recognise that computing has an impact on nearly every aspect of the world in which they live.

Computer Science - Year 11				
What are we learning?	**What knowledge, understanding and skills will we gain?**	**What does mastery look like?**	**How does this build on prior learning?**	**What additional resources are available?**
Computers, Hardware The internet and the world wide web	**Knowledge:** Physical storage devices (magnetic, optical, solid state) 'Cloud' and other contemporary secondary storage Embedded systems What is meant by the internet What is meant by the world wide web (WWW) Components of the WWW (web server URLs, ISP, HTTP, HTTPS, HTML) **Understanding:** Understand how data is stored on physical devices (magnetic, optical, solid state) Understand the concept of storing data in the 'cloud' and other contemporary secondary storage Understand the need for embedded systems and their functions Understand how the internet is structured (IP addressing, routers) Understand what is meant by the world wide web (WWW) and components of the WWW (web server URLs, ISP, HTTP, HTTPS, HTML)	Explain how data is stored on physical devices (magnetic, optical, solid state) Explain how data is stored on physical devices (magnetic, optical, solid state) Describe the concept of storing data in the 'cloud' and other contemporary secondary storage Describe the need for embedded systems and their functions Explain how the internet is structured (IP addressing, routers) Explain what is meant by the world wide web (WWW) and components of the WWW (web server URLs, ISP, HTTP, HTTPS, HTML)	Y10 Spring term 7.1 Y10 Summer term 5.1, 6.1,7.1,8.1	How networks across the globe are connected. Image here 'There and Back Again: A Packet's Tale – How does the Internet work?' https://www.youtube.com/watch?v=WwyJGzZmBe8 The difference between IP4 and IP6. Image of the output is here http://www.isaserver.org/img/upl/2004dh81115983842251.gif Layered protocol stacks to include MAC addresses. https://www.youtube.com/watch?v=uEe0rOcuVLo Visual trace route tool http://www.yougetsignal.com/tools/visual-tracert/ Network at http://teaching.shelswell.org.uk/wp-content/uploads/2009/12/roadnetwork.png Andrew Blum's TED lecture 'What is the internet really?' http://www.whois.sc/ 'How the Internet Works in 5 Minutes': http://www.youtube.com/watch?v=7_LPdttKXPc

	Skills: Explain how data is stored on physical devices (magnetic, optical, solid state) Describe the concept of storing data in the 'cloud' and other contemporary secondary storage Describe the need for embedded systems and their functions Explain how the internet is structured (IP addressing, routers) Explain what is meant by the world wide web (WWW) and components of the WWW (web server URLs, ISP, HTTP, HTTPS, HTML)			

Computer Science Unit Overview Year 11 NEA Analysis Stage 1 Term 1&2

Curriculum Unit Overview: Term 1, Term 2 NEA Stage 1 Analysis

The GCSE in Computer Science requires each student to undertake a programming project where they will develop a computer program. The project is not assessed and does not contribute towards the final grade. The purpose of the project is to enable students to develop skills in analysing the requirement of a problem, designing and then implementing a programming solution, testing, refining and evaluating their solution.

The project will require students to create a program that will include the following: data input and storage, processing data, producing output based on processed data.

Computer Science - Year 11				
What are we learning?	**What knowledge, understanding and skills will we gain?**	**What does mastery look like?**	**How does this build on prior learning?**	**What additional resources are available?**
Stage 1: Analysis	Knowledge: Know the requirements of the given brief Understanding: What the proposed solution will do to meet the requirements Identify the requirements of the program that will be designed, implemented and tested Skills: Identify the requirements of the problem Analyse the given problem Decompose the problem into manageable sub-problems	Report for analysis, the report should include: An introduction to the problem, in prose, which demonstrates an understanding of abstraction. Decompose the problem into manageable sub-problems, with an explanation of each. The decomposed list of requirements presented in prose or as a bulleted list, with each requirement clearly identified. A description of what each sub-problem will do is required, it can be presented in prose or as a bulleted list. An explanation, in prose, of the reasons why the decomposition submitted is the most appropriate to meet requirements must also be included.	Y10 problem solving and programming	NEA Delivery Guide Specification Pseudo-code booklet Initial test plan template Program Syntax Guide Data file

Computer Science Unit Overview Year 11 NEA Design Stage 2 Term 1&2

Curriculum Unit Overview: Term 1, Term 2 NEA Stage 2 Design

The GCSE in Computer Science requires each student to undertake a programming project where they will develop a computer program. The project is not assessed and does not contribute towards the final grade. The purpose of the project is to enable students to develop skills in analysing the requirement of a problem, designing and then implementing a programming solution, testing, refining and evaluating their solution.

The project will require students to create a program that will include the following: data input and storage, processing data, producing output based on processed data.

Computer Science- Year 11				
What are we learning?	**What knowledge, understanding and skills will we gain?**	**What does mastery look like?**	**How does this build on prior learning?**	**What additional resources are available?**
Stage 2: Design Test strategy and initial test plan	Knowledge: What is an algorithm What is a test strategy Understanding: How to design an algorithm Skills: Describe what has to be done when implementing the solution and to suggest an appropriate strategy to test the solution. Solution design, an algorithm or algorithms should be designed that meet/s the requirements of the problem using appropriate conventions (flowchart, pseudocode, written description). Devise a test strategy based on meeting the requirements of the problem	A written report including: the algorithm(s) any refinements to the design identified during implementation, with reasons for the refinements. Produce a test strategy and an initial test plan as below Test no | Purpose of the test | Test data | Expected result	Y10 problem solving and programming	NEA Delivery Guide Specification Pseudo-code booklet Initial test plan template Program Syntax Guide Data file

Computer Science Unit Overview Year 11 NEA Implementation Stage 3 Term 1&2

Curriculum Unit Overview: Term 1, Term 2 NEA Stage 3 Implementation

The GCSE in Computer Science requires each student to undertake a programming project where they will develop a computer program. The project is not assessed and does not contribute towards the final grade. The purpose of the project is to enable students to develop skills in analysing the requirement of a problem, designing and then implementing a programming solution, testing, refining and evaluating their solution.

The project will require students to create a program that will include the following: data input and storage, processing data, producing output based on processed data.

Computer Science- Year 11				
What are we learning?	**What knowledge, understanding and skills will we gain?**	**What does mastery look like?**	**How does this build on prior learning?**	**What additional resources are available?**
Stage 3: Implementation	Knowledge: Code required to build the programmed solution Understanding: How to program the solution to the problem Skills: Debugging Identify amendments to the original design solution which may become apparent during this stage Refinements should be implemented and documented as additions to the design and in the program code by using comments as descriptors.	Produce a report for implementation The report should include: a copy of the program code; any refinements should be noted as comments in the final program. Screenshots demonstrating effective use of debugging skills to correct errors.	Y10 problem solving and programming	NEA Delivery Guide Specification Pseudo-code booklet Initial test plan template Program Syntax Guide Data file

Computer Science Unit Overview Year 11 NEA Testing, Refining and Evaluation Stage 4 Term 1&2

Curriculum Unit Overview: Term 1, Term 2 NEA Stage 4 Testing, Refining and Evaluation

The GCSE in Computer Science requires each student to undertake a programming project where they will develop a computer program. The project is not assessed and does not contribute towards the final grade. The purpose of the project is to enable students to develop skills in analysing the requirement of a problem, designing and then implementing a programming solution, testing, refining and evaluating their solution.

The project will require students to create a program that will include the following: data input and storage, processing data, producing output based on processed data.

Computer Science - Year 11

What are we learning?	What knowledge, understanding and skills will we gain?	What does mastery look like?	How does this build on prior learning?	What additional resources are available?
Stage 4: Testing , Refining and Evaluation	Knowledge: How the final program relates to the brief Understanding: Show that the final program solution has been tested along with any refinements Skills: Evaluate solutions against the original requirements Attempt to correct and retest all errors	Produce a report which should include: The updated and complete Test Plan (labelled 'Final Test Plan') The evaluation Test no | Purpose of the test | Test data | Expected result | Actual result | Action needed/comments	Y10 problem solving and programming	NEA Delivery Guide Specification Pseudo-code booklet Initial test plan template Program Syntax Guide Data file

Computer Science Unit Overview Year 11 Term 2.2

Computers are able to store and manipulate large quantities of data. They use binary to represent different types of data. Students are expected to learn how different types of data are represented in a computer.

Computer Science - Year 11				
What are we learning?	**What knowledge, understanding and skills will we gain?**	**What does mastery look like?**	**How does this build on prior learning?**	**What additional resources are available?**

Data: Databases Characteristics of structured and unstructured data Structured databases Tables and relationships Encryption: Caesar cipher algorithm	**Knowledge:** What is structured and unstructured data Know that data can be decomposed, organised and managed What are tables, records, fields, relationships, keys What is data encryption Caesar cipher Understanding: Understand the characteristics of structured and unstructured data understand that data can be decomposed, organised and managed in a structured database Understand the need for data encryption Understand how a Caesar cipher algorithm works Skills: Analyse a flat-file database Devise a cipher Encrypting and decrypting using a Caesar cipher wheel.	Be able to analyse a flat-file database Be able to devise a cipher Be able to encrypt and decrypt using a Caesar cipher wheel.	Y10 Spring lesson 11.1, 12.1 Y10 Summer lesson 1.1,1.2,3.1,4.1	YouTube video '158,962,555,217,826,360,000 (Enigma Machine) - Numberphile' to introduce the Enigma machine: http://www.youtube.com/watch?v=G2_Q9FoD-oQ Ciphertext challenge: http://simonsingh.net/cryptography/cipher-challenge/the-ciphertexts/stage-1/ 'Database terms': http://www.youtube.com/watch?v=Kwcs-BwLFmg YouTube video 'Caesar Cipher': http://www.youtube.com/watch?v=sMOZf4GN3oc 'The Caesar Shift Wheel': http://www.maths-resources.net/enrich/codes/caesar/caesarwheel.pdf YouTube videos 'Database Key fields - animation' (https://www.youtube.com/watch?v=JkwbhFUftSc) and 'DatabaseKeysrelationship' (https://www.youtube.com/watch?v=djqgCzZem7w) TEDxWarwick – Andy Standford-Clark – Innovation Begins at Home (https://www.youtube.com/watch?v=s9nrm8q5eGg) Tim Berners-Lee: The next Web of open, linked data (https://www.youtube.com/watch?v=OM6XIICm_qo)