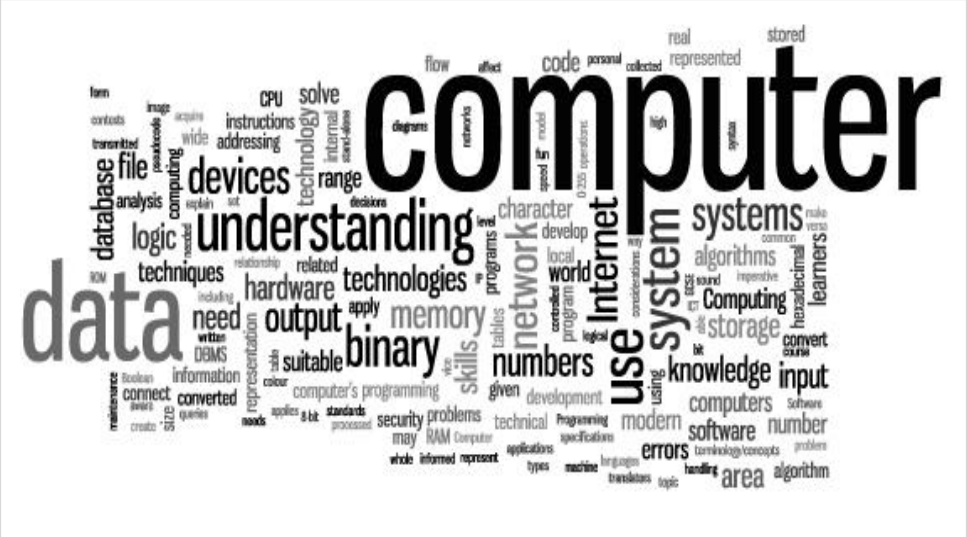
PERSONAL LEARNER CHECKLIST
KS4

GCSE Computer Science



Curriculum Information:
Computer Science

Subject: Computer Science

Subject Leader: Mr K. Lau-Thomas **Email address:** klau@christtheking.notts.sch.uk

What Specification (syllabus) is being taught?	OCR GCSE Computer Science (9-1) from 2015
What are the key topics and themes? When will they be taught?	Component 01 - Computer Systems Component 02 – Computational thinking, algorithms and Programming Component 03 – Programming project
How will my son or daughter be assessed? When do these assessments take place?	Component 01 – 1 hour and half exam worth 50% May 2018 Component 02 – 1 hour and half exam worth 50% may 2018
What can my son or daughter do for revision at home? What materials are provided or available online?	-Use their Computing revision guides and text books that they have been issued with from school to help structure their revision. - Make use of their classwork booklets and exam papers. - Use of the below websites. - Highly recommended resources that is mapped to the specification and allows tracking of progress through the course. www.senecalearning.com -Select KS4 computing for the Theory units covered in the specification. http://www.teach-ict.com/ -Good Computer Science theory. Caution some of the content is for the new computer Science specification. http://www.bbc.co.uk/education/subjects/z34k7ty -Excellent resources mapped directly to the OCR GCSE Computing Specification. http://www.cambridgegcsecomputing.org/ -On YouTube search “Craig n Dave” and access the OCR tutor videos for extensive explanation on Computer Science topics -On the app store search for OCR Past Exam Papers good source of past papers. -On Android search for GCSE Computing Revision by Mr S Whorton. Good computing theory with questions.

Component 01 – Computer Systems

Systems Architecture				
1	The purpose of the CPU			
2	Von Neumann architecture: MAR (Memory Address Register) , MDR (Memory Data Register) Program Counter , Accumulator			
3	Common CPU components and their function: ALU (Arithmetic Logic Unit) , CU (Control Unit), Cache			
4	The function of the CPU as fetch and execute instructions stored in memory			
5	How common characteristics of CPUs affect their performance: clock speed , cache size, number of cores			
6	Embedded systems: purpose of embedded systems , examples of embedded systems			

Memory				
1	The difference between RAM and ROM			
2	The purpose of ROM in a computer system			
3	The purpose of RAM in a computer system			
4	The need for virtual memory			
5	Flash memory			

Storage				
1	The need for secondary storage			
2	Data capacity and calculation of data capacity requirements			
3	Common types of storage: optical , magnetic , solid state			
4	Suitable storage devices and storage media for a given application, and the advantages and disadvantages of these, using characteristics: capacity , speed , portability , durability, reliability, cost			
5	The need for virtual memory			
6	Cache memory			
7	Flash memory			
8	How changes in memory technologies are leading to innovative computer designs			

Wired and Wireless networks				
1	Types of networks: LAN (Local Area Network), WAN (Wide Area Network)			
2	Factors that affect the performance of networks			
3	The different roles of computers in a client-server and a peer-to-peer network			
4	The hardware needed to connect stand-alone computers into a Local Area Network: wireless access points , routers/switches , NIC (Network Interface Controller/Card) , transmission media			
5	The internet as a worldwide collection of computer networks: DNS (Domain Name Server) , Hosting, the cloud			
6	The concept of virtual networks			

Network Topologies, Protocols and Layers				
1	Star and mesh network topologies			
2	Wifi: frequency and channels , encryption			
3	Ethernet			
4	The uses of IP addressing, MAC addressing, and protocols including: TCP/IP (Transmission Control Protocol/Internet Protocol) , HTTP (Hyper Text Transfer Protocol), HTTPS (Hyper Text Transfer Protocol Secure) , FTP (File Transfer Protocol) , POP (Post Office Protocol) IMAP (Internet Message Access Protocol) , SMTP (Simple Mail Transfer Protocol)			
5	The concept of layers			
6	Packet switching			

Systems Security				
1	Forms of attack			
2	Threats posed to networks: malware , phishing , people as the 'weak point' in secure systems (social engineering), brute force attacks denial of service attacks, data interception and theft, the concept of SQL injection, poor network policy			
3	Identifying and preventing vulnerabilities: penetration testing, network forensics, network policies, anti-malware software, firewalls user access levels, passwords, encryption			

Systems Software				
1	The purpose and functionality of systems software			
2	Operating systems: user interface, memory management/multitasking peripheral management and drivers, user management, file management			
3	Utility system software: encryption software, defragmentation, data compression			
4	The role and methods of backup: full, incremental			

Ethical, Legal, Cultural and Environmental concerns				
1	How to investigate and discuss Computer Science technologies while considering: ethical issues, legal issues, cultural issues, environmental issues, privacy issues			
2	How key stakeholders are affected by technologies			
3	Environmental impact of Computer Science			
4	Cultural implications of Computer Science			
5	Open source vs proprietary software			
6	Legislation relevant to Computer Science: The Data Protection Act 1998 , Computer Misuse Act 1990 , Copyright Designs and Patents Act 1988 , Creative Commons Licensing , Freedom of Information Act 2000			

Component 02 - Computational thinking, algorithms and Programming

Algorithms				
1	Computational thinking: abstraction , decomposition , algorithmic thinking			
2	Standard searching algorithms: binary search , linear search			
3	Standard sorting algorithms: bubble sort , merge sort , insertion sort			
4	How to produce algorithms using: pseudocode , using flow diagrams			
5	Interpret, correct or complete algorithms			

Programming Techniques				
1	The use of variables, constants, operators, inputs, outputs and assignments			
2	The use of the three basic programming constructs used to control the flow of a program: sequence , selection iteration (count and condition controlled loops)			
3	The use of basic string manipulation			
4	The use of basic file handling operations: open , read write , close			
5	The use of records to store data			
6	The use of SQL to search for data			
7	The use of arrays (or equivalent) when solving problems, including both one and two dimensional arrays			
8	How to use sub programs (functions and procedures) to produce structured code			
9	The use of data types: integer , real , boolean , character and string , casting			
10	The common arithmetic operators			
11	The common Boolean operators			

Producing robust programs				
1	Defensive design considerations: input sanitisation/validation , planning for contingencies , anticipating misuse , authenticating			
2	Maintainability: comments , indentation			
3	The purpose of testing			
4	Types of testing: iterative , final/terminal			
5	How to identify syntax and logic errors			
6	Selecting and using suitable test data			

Computational Logic				
1	Why data is represented in computer systems in binary form			
2	Simple logic diagrams using the operations AND, OR and NOT			
3	Truth tables			
4	Combining Boolean operators using AND, OR and NOT to two levels			
5	Applying logical operators in appropriate truth tables to solve problems			
6	Applying computing-related mathematics: + , - , / , * , Exponentiation (^) , MOD , DIV			

Translators and Facilities of the Language				
1	Characteristics and purpose of different levels of programming language, including low level languages			
2	The purpose of translators			
3	The characteristics of an assembler, a compiler and an interpreter			
4	Common tools and facilities available in an integrated development environment (IDE): editors, error diagnostics, run-time environment, translators.			

Data Representation				
1	**Units** • bit, nibble, byte, kilobyte, megabyte, gigabyte, terabyte, petabyte			
2	• how data needs to be converted into a binary format to be processed by a computer.			
3	**Numbers** • how to convert positive denary whole numbers (0–255) into 8 bit binary numbers and vice versa			
4	• how to add two 8 bit binary integers and explain overflow errors which may occur			
5	• binary shifts			
6	• how to convert positive denary whole numbers (0–255) into 2 digit hexadecimal numbers and vice versa			
7	• how to convert from binary to hexadecimal equivalents and vice versa			
8	• check digits			
9	**Characters** • the use of binary codes to represent characters			
10	• the term ‘character-set’			
11	• the relationship between the number of bits per character in a character set and the number of characters which can be represented (for example ASCII, extended ASCII and Unicode)			
12	**Images** • how an image is represented as a series of pixels represented in binary.			
13	• metadata included in the file			
14	• the effect of colour depth and resolution on the size of an image file			
15	**Sound** • how sound can be sampled and stored in digital form			
16	• how sampling intervals and other factors affect the size of a sound file and the quality of its playback: sample size, bit rate, sampling frequency			
17	**Compression** • need for compression			
18	• types of compression: lossy, lossless			