GCSE Computer Science

Curriculum Intent 2021-2022

Computer Science is a subject in demand within in a globally competitive world. It has become an ever-growing part of human life, affecting many aspects of a person's day. Computer systems are embedded ubiquitously in everyday devices, smart phones, washing machines, heating systems and vehicles, as our world embraces "The Internet of Things". Computer scientists have an impact on how our society advances by developing and maintaining these systems: whether it be for our home, work, learning or entertainment environments. Computer Science is an exciting and rapidly evolving subject that offers excellent employment prospects and well-paid careers.

The Computer Science department at Brine Leas aims to develop the mind-set of a computer scientist through equipping students with the skills to participate in a rapidly-changing world. The curriculum journey incorporates challenging and engaging topics, giving students the opportunity to develop their capability, creativity and subject knowledge. It also capitalises upon, and feeds into, learning within other subject disciplines including mathematics, science and design and technology. It is these opportunities that enable students to develop and hone skills that can be applied in day-to-day life.

Steve Jobs said "Everyone should learn how to code, it teaches you to think." Computational thinking is an essential skill for everyone to have and helps in all subject areas and careers.

The curriculum has been designed to ensure learners have sufficient knowledge to stay safe online and use computers safely in life. We want students to not only understand how to use technology effectively, safely and responsibly, but also how technology is developed and constantly redeveloped into new and exciting tools. The curriculum also provides a focus on developing resilient learners who are able to recover from mistakes and effectively solve problems. This will help develop a lifelong effect of learning and how to develop themselves further and prepare for the future.

The curriculum is developed so that students are taught the principles of problem solving and computation, which prepares them to solve the problems of tomorrow, by developing learner's knowledge, skills and understanding through key computational concepts and experience. They develop understanding for all the technology that surrounds them by not just understanding how computer systems work, but how to put this knowledge to use through programming and problem solving. Building on this knowledge and understanding, students are equipped to use information technology to create programs, systems and a range of content whilst developing programming skills. Students will also analyse problems in computational terms and devise creative solutions by designing, writing, testing and evaluating programs. This also ensures that students become digitally literate – able to use, and express themselves and develop their ideas through, information technology – at a level suitable for the future workplace and as active participants in a digital world. We endeavour to make the curriculum as

motivational and interesting as possible with a high level of challenge by offering breadth and depth of experiences for the students. Our aim is to ensure that students develop and achieve ICT capability that is directly transferable, not only to other subjects, but also to the future learning pathways and beyond, developing a wide range of digital skills that will prepare learners for the future.

The key stage 3 curriculum provides challenges and new experiences in computing, digital literacy and digital media (regardless of their prior knowledge of using computers) and is designed to ensure students studying GCSE Computer Science have a basis of knowledge, skills and understanding in the fundamental concepts covered at KS4. Over time, students learn to, and develop proficiency in, program in 3 languages, starting with block-based languages before progressing to High-Level Languages. The development of programming skills is also built into physical Computing tasks using Micro:Bits for example coding LED lights to effectively apply the knowledge learnt in earlier Algorithm and Programming units. The curriculum journey connects to other curriculum areas holistically to ensure learning contexts are authentic, meaningful and provide opportunities for application of skills, investigation and purposeful play. In addition, references to key events and developments through the history of technology using role models from all aspects of society are used in an inspirational and motivational way for students.

We aim to enable students to develop a love for the subject and an understanding that there are no limits to their own development in programming and IT. To enthuse students to have an understanding far deeper than the interface that they currently operate. This is done by offering challenging opportunities and personal development.

A wealth of irresistible and enriching learning opportunities are open to all students to enrich educational experiences, to engage learners, and cultivate and extend lifelong effect of learning. All students in year 9 will take part in challenging opportunities by completing The Bebras challenge and achieve badges for their iDEA award. Girls in year 8 compete in the CyberFirst Girls competition. The CyberFirst Girls Competition provides a fun and challenging environment to inspire the next generation of young women to consider a career in cyber security. Furthermore, students are given the opportunity to enter a range of National Competitions such as game design and development for YGD BAFTA and competing against other secondary schools in CyberFirst and Cyber Centurion events. The CyberFirst Competition provides a fun and challenging environment to inspire the next generation of young people to consider a career in cyber security.

Due to the forever changing world of technology the curriculum and skills need is taken into account. Staff are involved with the local primary schools and the whole community including Computing at School and exam boards to ensure that the curriculum is achievable and forward thinking, to ensure that students are equipped for their future pathways.

Assessment

Please see website for the formal internal assessment record.
Formal external assessment:
Paper 1 Written exam, subject coverage:

    - Fundamentals of algorithms
    - Programming
    - Fundamentals of data representation
    - Computer systems
- Written exam set in practically based scenarios: 1 hour 30 minutes
- 80 marks

Paper 2 Written exam, subject coverage:

    - Fundamentals of data representation
    - Computer systems
    - Fundamentals of computer networks
    - Fundamentals of cyber security
    - Ethical, legal and environmental impacts of digital technology on wider society, including issues of privacy
- Written exam: 1 hour 30 minutes
- 80 marks
- 50% of GCSE each.

Homework

Axsied booklet/Repl.it set homework for coding and theory practice

Clubs and/or intervention

Lunchtime drop-in sessions available. Extra revision available after school as needed.

Parental/Carer support

Parents/Carers can find the subject content and specification at: https://filestore.aqa.org.uk/resources/computing/specifications/AQA-8520-SP-2016.PDF.

Helpful sources of information

https://www.aqa.org.uk/subjects/computer-science-and-it/gcse/computer-science-8520 - exam board subject homepage
https://www.senecalearning.com/ - This is a highly effective revision website that tracks pupil progress through the specification
https://www.bbc.com/bitesize/subjects/z34k7ty – BBC Bitesize revision theory and questions
https://repl.it – programming practice challenges
https://w3schools.com – tutorials, references for programming languages
www.codeacademy.com – learn technical skills in an interactive environment
www.teach-ict.com

Connections to future pathways

Careers - Jobs directly related: Application analyst, Applications developer, CAD technician, Cyber security analyst, Data analyst, Database administrator, Forensic computer analyst, Game designer, Games developer, Information systems manager, IT consultant, Machine learning engineer, Multimedia programmer, Penetration tester, SEO specialist, Software engineer, Systems analyst, UX designer, VFX artist, Web designer, Web developer
Jobs where your degree would be useful include: IT sales professional, IT trainer, Nanotechnologist, Network engineer, Supply chain manager, Telecommunications researcher

Future learning: A Level Computer Science, A Level Mathematics, AQA Tech Level IT-Programming

Significant local employers include: Barclays (Knutsford),

Year 10 Overview

Term	Knowledge	Assessment	Connections to learning
Autumn 1	**Algorithms** Can you solve puzzles and problems? How do you solve problems? We will show you how to solve a problem using either pseudo-code or a flowchart (using symbols learnt in Key stage 3). How good will you be to solve a problem without a computer? Students need a theoretical understanding of programming code even if they are not using a programming language. Having good, strong problem solving skills can make a huge difference to your career. Problems are at the centre of what many people do at work every day. Written exams will always present algorithms and code segments.		
	➢ Algorithms ➢ Algorithms are used to solve problems ➢ Decomposition and abstraction are used to simplify problems. ➢ Advantages of a systematic approach for problem solving and algorithm creation. ➢ Pseudo-code and flowcharts are used to solve algorithms.	➢ Program challenges ➢ Teacher/pupil questioning ➢ Exam style question practice (homework's and in class) ➢ Axsied booklet ➢ PGOnline resources	➢ Basics of all computer programming ➢ Links to AQA A Level Computer Science 4.4.1.1 Problem Solving 4.4.1.3 Abstraction ➢ Resilience
	➢ Algorithms can be written in various formats to solve the same problem. ➢ Advantages and disadvantages of different algorithms.	➢ Program challenges ➢ Teacher/pupil questioning ➢ Exam style question practice (homework's and in class) ➢ Axsied booklet ➢ PGOnline resources	➢ Links to AQA A Level Computer Science ➢ 4.4.1.1 Problem solving

		➢	
	➢ The process of a linear and binary search. ➢ Compare and contrast linear and binary search with algorithms.	➢ Program challenges ➢ Teacher/pupil questioning ➢ Exam style question practice (homework's and in class) ➢ Axsied booklet ➢ PGOnline resources	➢ Links to AQA A Level Computer Science 4.3.4 searching algorithms 4.3.4.1 Linear search ➢ 4.3.4.2 Binary search
	➢ The process of merge and bubble sort. ➢ Compare and contrast merge sort and bubble sort algorithms.	➢ Program challenges ➢ Teacher/pupil questioning ➢ Exam style question practice (homework's and in class) ➢ Axsied booklet ➢ PGOnline resources	➢ Links to AQA A Level Computer Science 4.3.5 sorting algorithms 4.3.5.1 Bubble sort ➢ 4.3.5.2 Merge sort
	➢ Number bases: • decimal (base 10) • binary (base 2) • hexadecimal (base 16). ➢ Computers use binary to represent all data and instructions. ➢ Hexadecimal is often used in computer science.	➢ Teacher/pupil questioning ➢ Exam style question practice (homework's and in class) ➢ Axsied booklet	➢ Links to AQA A Level Computer Science 4.5.2 Number bases ➢ 4.5.2.1 Number base
	➢ Binary can be used to represent whole numbers. ➢ Hexadecimal can be used to represent whole numbers. ➢ Binary and hexadecimal can convert in both directions between: • binary and decimal • binary and hexadecimal • decimal and hexadecimal.	➢ Teacher/pupil questioning ➢ Exam style question practice (homework's and in class) ➢ Axsied booklet	➢ Links to AQA A Level Computer Science ➢ 4.5.2.1 Number base
	➢ A bit is the fundamental unit of information ➢ A byte is a group of 8 bits.	➢ Teacher/pupil questioning ➢ Exam style question practice (homework's and in class)	➢ Links to AQA A Level Computer Science 4.5.3.1 Bits and bytes

	➢ Quantities of bytes can be described using prefixes. ➢ Decimal Prefixes have names, symbols and corresponding Values: • kilo, 1 kB is 1,000 bytes • mega, 1 MB is 1,000 kilobytes • giga, 1 GB is 1,000 Megabytes • tera, 1 TB is 1,000 Gigabytes.	➢ Axsied booklet	➢ 4.5.3.2 Units
	➢ Binary numbers can be added together. ➢ Binary shifts can be applied to a binary number to indicate multiplication and division.	➢ Teacher/pupil questioning ➢ Exam style question practice (homework's and in class) ➢ Axsied booklet	➢ Links to AQA A Level Computer Science ➢ 4.5.4.2.Unsigned binary arithmetic
	➢ Character set is a binary bit pattern that represents character in coding methods. ➢ 7-bit ASCII is a type of character set. ➢ Character codes are commonly grouped and run in sequence within encoding tables. ➢ Unicode is a character when there are not enough characters for other languages. ➢ The advantages of Unicode over ASCII. ➢ Unicode uses the same codes as ASCII up to 127.	➢ Teacher/pupil questioning ➢ Exam style question practice (homework's and in class) ➢ Axsied booklet	➢ Links to AQA A Level Computer Science ➢ 4.5.5.1 character form of a decimal number
	➢ Pixels relate to an image and the way images are displayed. ➢ Bitmaps: • size in pixels • colour depth.	➢ Teacher/pupil questioning ➢ Exam style question practice (homework's and in class) ➢ Axsied booklet	➢ Links to AQA A Level Computer Science 4.5.6.1 bit pattern, images, sound and other data 4.5.6.2 Analogue and digital 4.5.6.3 Analogue/digital conversion

	➢ Image resolution size of a bitmap image in pixels (width x height). ➢ Bitmap represents an image using pixels and colour depth. ➢ Number of pixels and colour depth can affect the file size of a bitmap image. ➢ Calculate bitmap image file sizes ➢ Convert binary data into a black and white image. ➢ Convert a black and white image into binary data.		4.5.6.4 Bitmapped graphics 4.5.6.5 Vector graphics ➢ 4.5.6.6. Vector graphics versus bitmapped graphics
Autumn 2	**Programming concepts** Can you identify the need for different data types? Can you add 'n' to 'n'? Understanding the need to have different data types in theory will help when using text based programming languages. Resilience is something that you will develop in the section and is an important life skills along with perseverance to create a solution even when things go wrong.		
	➢ Concept of a data types: • integer • real • Boolean • character • string	➢ Program challenges ➢ Teacher/pupil questioning ➢ Exam style question practice (homework's and in class) ➢ Axsied booklet ➢ PGOnline resources	➢ Links to AQA A Level Computer Science 4.1.1.1 Data types
	➢ variable declaration ➢ constant declaration ➢ assignment ➢ iteration ➢ selection ➢ subroutine(procedure/function). ➢ Definite and indefinite iteration, including indefinite iteration with the condition(s) at the start or the end of the iterative structure. ➢ Nested selection and nested iteration structures. ➢ Meaningful identifier names and know why it is important to use them.	➢ Program challenges ➢ Teacher/pupil questioning ➢ Exam style question practice (homework's and in class) ➢ Axsied booklet ➢ PGOnline resources	➢ Links to AQA A Level Computer Science 4.1.1.2 programming concepts 4.1.1.6 constants & variables in a programming language 4.4.1.2 following & writing algorithms

	➢ addition ➢ subtraction ➢ multiplication ➢ real division ➢ integer division, including remainders.	➢ Program challenges ➢ Teacher/pupil questioning ➢ PGOnline resources	➢ Mathematics KS2 simple calculations ➢ Links to AQA A Level Computer Science 4.1.1.3 arithmetic operations in a programming language
	➢ equal to ➢ not equal to ➢ less than ➢ greater than ➢ less than or equal to ➢ greater than or equal to.	➢ Program challenges ➢ Teacher/pupil questioning ➢ Exam style question practice (homework's and in class) ➢ PGOnline resources	➢ Mathematics KS2 simple calculations ➢ Links to AQA A Level Computer Science 4.1.1.4 relational operations in a programming language
	➢ NOT ➢ AND ➢ OR	➢ Program challenges ➢ Teacher/pupil questioning ➢ Exam style question practice (homework's and in class)	➢ Links to AQA A Level Computer Science 4.1.1.5 Boolean operation in a programming language ➢ Mathematics ➢ Electronics
	➢ Concept of data structures. ➢ Arrays (or equivalent) in the design of solutions to simple problems. ➢ Records (or equivalent) in the design of solutions to simple problems.	➢ Program challenges ➢ Teacher/pupil questioning ➢ Exam style question practice (homework's and in class) ➢ Axsied booklet ➢ PGOnline resources	➢ Links to AQA A Level Computer Science 4.10 Fundamentals of databases
	➢ Sound is analogue and that it must be converted to a digital form for storage and processing in a computer. ➢ Sound waves are sampled to create the digital version of sound. ➢ Digital representation of sound in terms of: • sampling rate • sample resolution	➢ Multiple Choice Question Activities ➢ Teacher/pupil questioning ➢ Exam style question practice (homework's and in class) ➢ Axsied booklet	➢ Links to AQA A Level Computer Science 4.5.6.7 Digital representation of sound 4.5.6.8. Musical Instrument Digital Interface (MIDI)

	➢ Calculate sound file sizes based on the sampling rate and the sample resolution.		
	➢ Data compression and why data may be compressed and that there are different ways to compress data. ➢ Data can be compressed using Huffman coding. ➢ Interpret/create Huffman trees. ➢ Calculate the number of bits required to store a piece of data compressed using Huffman coding. ➢ Calculate the number of bits required to store a piece of uncompressed data in ASCII. ➢ Data can be compressed using run length encoding (RLE) and represent data in RLE frequency/data pairs.	➢ Multiple Choice Question Activities ➢ Teacher/pupil questioning ➢ Exam style question practice (homework's and in class) ➢ Axsied booklet	➢ Links to AQA A Level Computer Science 4.5.6.9 Data compression
Spring 1	**Hardware and software** Can you create a truth table for a given logic gate? Understand how logic gates are used as the basis for most computer components. Know why different types of software are needed.		
	➢ Obtain user input from the keyboard. ➢ Output data and information from a program to the computer display. ➢ Read/write from/to a text file.	➢ Program challenges ➢ Teacher/pupil questioning ➢ PGOnline resources	➢ Links to AQA A Level Computer Science
	➢ In programming use: • Length • Position • Substring • Concatenate • convert character to character code • convert character code to character • string conversion operations.	➢ Program challenges ➢ Teacher/pupil questioning ➢ PGOnline resources	➢ Links to AQA A Level Computer Science 4.1.1.7 string-handling operations in a programming language

	➢ Use random number generation.	➢ Program challenges ➢ Teacher/pupil questioning ➢ PGOnline resources	➢ Links to AQA A Level Computer Science 4.1.1.8 random number generation in a programming language
	➢ Hardware and software and the relationship between them.	➢ Teacher/pupil questioning	➢ Links to AQA A Level Computer Science 4.6.1 Hardware and software
	➢ Construct truth tables for the following logic gates: • NOT • AND • OR. ➢ Construct truth tables for simple logic circuits. Interpret the results of simple truth tables. ➢ Create, modify and interpret simple logic circuit diagrams.	➢ Multiple Choice Question Activities ➢ Teacher/pupil questioning ➢ Exam style question practice (homework's and in class) ➢ Axsied booklet	➢ Links to AQA A Level Computer Science 4.6.4 Logic gates ➢ Electronics ➢ Mathematics
	➢ What is meant by: • system software • application software. ➢ Functions of, operating systems (OS) and utility programs. ➢ OS handles management of the: • processor(s) • memory • I/O devices • applications • security.	➢ Multiple Choice Question Activities ➢ Teacher/pupil questioning ➢ Exam style question practice (homework's and in class) ➢ Axsied booklet	➢ Links to AQA A Level Computer Science 4.6.1.2. Classification of software 4.6.1.3. System software 4.6.1.4 Role of an operating system (OS)
	➢ Von Neumann architecture. ➢ The role and operation of main memory and the following major components of a central processing unit (CPU): • arithmetic logic unit • control unit • clock	➢ Multiple Choice Question Activities ➢ Teacher/pupil questioning ➢ Exam style question practice (homework's and in class) ➢ Axsied booklet	➢ Links to AQA A Level Computer Science 4.7.1 Internal hardware components of a computer 4.7.1.1 Internal hardware components of a computer 4.7.2. The stored program concept

	• bus ➢ The effect of the following on the performance of the CPU: • clock speed • number of processor cores • cache size • cache type ➢ The Fetch-Execute cycle. ➢ Differences between main memory and secondary storage. ➢ Differences between RAM and ROM. ➢ Secondary storage and why it is required. ➢ Different types of secondary storage (solid state, optical and magnetic). ➢ Operation of solid state, optical and magnetic storage and the advantages of each. ➢ 'Cloud storage' and the advantages and disadvantages. ➢ 'Embedded systems' and how the system differs from a non-embedded system.		4.7.2.1. The meaning of the store program concept 4.7.3 Structure and role of the processor and its components 4.7.3.1 The processor and its components 4.7.3.2 The fetch-execute cycle and the role of registers within it 4.7.4.1 Input and output devices 4.7.4.2 Secondary storage devices
Spring 2	**Networking** Do you know what type of network is used in school? How do you know if you are using a safe/secure network to store/save/access your documents? Understand the basic architecture behind main computer networks. Know why different topologies suit different scenarios.		
	➢ Concept of subroutines and the advantages of using subroutines in programs. ➢ Parameters to pass data within programs and their use. ➢ Subroutines that return values to the calling routine. ➢ Local variables may be declared in subroutines:	➢ Program challenges ➢ Teacher/pupil questioning	➢ Links to AQA A Level Computer Science 4.1.1.10 subroutines (procedures/functions) 4.1.1.11 Parameters of subroutines 4.1.1.12 returning value/values from a subroutine 4.1.1.13 local variables in a subroutine

	• only exist while the subroutine is executing • are only accessible within the subroutine. ➢ Use local variables and why it is good practice to do so.		4.1.1.14 global variables in a programming language
	➢ Structured approach to programming and the advantages of the structured approach.	➢ Multiple Choice Question Activities ➢ Teacher/pupil questioning ➢ Exam style question practice (homework's and in class)	➢ Links to AQA A Level Computer Science 4.1.2.2 procedural-orientated programming
	➢ Write simple data validation routines. ➢ Authentication routines. ➢ Select suitable test data that covers normal (typical), boundary (extreme) and erroneous data. ➢ Justify the choice of test data.	➢ Multiple Choice Question Activities ➢ Teacher/pupil questioning	➢ Links to AQA A Level Computer Science
	➢ What a computer network is. ➢ Benefits and risks of computer networks. ➢ Main types of computer network including: • Personal Area Network (PAN) • Local Area Network (LAN) • Wide Area Network (WAN). ➢ Networks can be wired or wireless. ➢ Benefits and risks of wireless networks as opposed to wired networks. ➢ Common network topologies: • star • bus. ➢ 'Network protocol'. ➢ Purpose and use of common network protocols ➢ The need for, and importance of, network security. ➢ Methods of network security:	➢ Multiple Choice Question Activities ➢ Teacher/pupil questioning ➢ Exam style question practice (homework's and in class) ➢ Axsied booklet	➢ Links to AQA A Level Computer Science 4.9.2.1 Network topology 4.9.2.2 Types of networking between hosts 4.9.2.3 Wireless networking 4.9.4.1 TCP/IP 4.9.4.2 Standard application layer protocols

	• authentication • encryption • firewall • MAC address filtering. ➢ 4 layer TCP/IP model: • application layer • transport layer • internet layer • link layer. ➢ TCP and UDP are protocols that operate at the transport layer. ➢ IP protocol operates at the internet layer.		
Summer 1	**Low level/High Level languages** Why do we use text-based languages when computers only understand 1's and 0's? Understand the advantages of using high- and low-level languages when programming computers.		
	➢ Different levels of programming language: • low-level language • high-level language. ➢ Main differences between low-level and high-level languages. ➢ Machine code and assembly language are considered to be low-level languages and discuss the differences between them. ➢ Programming code written in high-level or assembly languages must be translated into machine code. ➢ Machine code is expressed in binary and is specific to a processor or family of processors. ➢ Advantages and disadvantages of low-level language programming compared with high-level language programming. ➢ Three common types of program translator:	➢ Multiple Choice Question Activities ➢ Teacher/pupil questioning ➢ Exam style question practice (homework's and in class) ➢ Axsied booklet	➢ Links to AQA A Level Computer Science 4.3.1.1 types of program translator

	• interpreter • compiler • assembler. ➢ Differences between these three types of translator and when it would be appropriate to use each type.		
Summer 2	**Cyber security** Who do you share information with? Is the information safe? Why do we have to verify that we are not a robot when creating accounts and passwords? Understand the need for awareness in the ever-changing data rich environment that we use every day. How to mimise threats to you data and system. With the employment of information security analysts projected to grow 28 percent from 2016 to 2026, much faster than the average for all other occupations. Demand for information security analysts is expected to be very high as analysts will need to create innovative solutions to prevent malicious intruders from stealing critical information or causing problems for computer networks. This section with give you the insights into cyber security and what it entails.		
	➢ Cyber security and the main purposes.	➢ Multiple Choice Question Activities ➢ Teacher/pupil questioning ➢ Exam style question practice (homework's and in class)	➢ Links to AQA A Level Computer Science 4.9.3.2 Internet security
	➢ Cyber security threats: • social engineering techniques • malicious code • weak and default passwords • misconfigured access rights • removable media • unpatched and/or outdated software. ➢ Penetration testing and what it is used for.	➢ Multiple Choice Question Activities ➢ Teacher/pupil questioning ➢ Exam style question practice (homework's and in class) ➢ Axsied booklet	➢ Links to AQA A Level Computer Science 4.9.3.2 Internet security

	➢ Social engineering. ➢ Social engineering and how it can be protected against. ➢ Social engineering: • blagging (pretexting) • phishing • pharming • shouldering (or shoulder surfing).	➢ Multiple Choice Question Activities ➢ Teacher/pupil questioning ➢ Exam style question practice (homework's and in class) ➢ Axsied booklet	➢ Links to AQA A Level Computer Science 4.9.3.2 Internet security
	➢ 'Malware'. ➢ How it can be protected against. ➢ Forms of malware: • computer virus • trojan • spyware • adware.	➢ Multiple Choice Question Activities ➢ Teacher/pupil questioning ➢ Exam style question practice (homework's and in class)	➢ Links to AQA A Level Computer Science 4.9.3.2 Internet security
	➢ Security measures: • biometric measures (particularly for mobile devices) • password systems • CAPTCHA (or similar) • using email confirmations to confirm a user's identity • automatic software updates.	➢ Multiple Choice Question Activities ➢ Teacher/pupil questioning ➢ Exam style question practice (homework's and in class) ➢ Axsied booklet	➢ Links to AQA A Level Computer Science 4.9.3.2 Internet security

Year 11 Overview

Term	Knowledge	Assessment	Representations
Autumn 1	**Relational databases and structured query language** Understand the fundamentals of databases. The concept of databases and how SQL is used within programming to access information.		

	➢ Databases ➢ Relational databases • table • record • field • primary key • foreign key	➢ Multiple Choice Question Activities ➢ Teacher/pupil questioning ➢ Exam style question practice (homework's and in class) ➢ Axsied booklet ➢ Application of knowledge understanding and skills using SQL	➢ Teacher demonstration and explanations ➢ Video clips
	➢ SQL to retrieve data from multiple tables of a relational database ➢ SQL commands: • SELECT • FROM • WHERE • ORDER BY…ASC | DESC ➢ insert, edit and delete data		
Autumn 2	**Ethical, legal and environmental impacts of digital technology on wider society, including issues of privacy** New technology, but who is collecting the data and who is listening, Alexa !! Hey Google !! Driverless cars is there is an accident who is at fault, the car, manufacturer or driver ?		
	➢ Current ethical, legal and environmental impacts and risks of digital technology on society. ➢ Software and their algorithms embed moral and cultural values ➢ Data privacy issues how they arise • cyber security • mobile technologies • wireless networking • cloud storage • hacking (unauthorised access to a computer system) • wearable technologies • computer based implants • autonomous vehicles.	➢ Multiple Choice Question Activities ➢ Teacher/pupil questioning ➢ Exam style question practice (homework's and in class) ➢ Axsied booklet	➢ Teacher demonstration and explanations ➢ Videos clips

Spring 1	**Preparing for the exams** Should we have drivels cars? Who would be at fault in an accident? Should we use technology and genetics? Should anybody be allowed to have/fly a drone? Understand how technology is developing and how we need to consider ethics as well and the need for developments in technology.		
	➢ Exam skills and misconceptions for Mock Exams ➢ practice exam papers (past & Exampro)	➢ Multiple Choice Question Activities ➢ Teacher/pupil questioning ➢ Exam style question practice (homework's and in class)	➢ Teacher demonstration and explanations ➢ Student example answers ➢ Key terms and definitions ➢ Exam walk throughs.
Spring 2	**Preparing for the exams** You will be using the lessons to look at exam techniques, go through past papers, revising different topics and reinforcing your learning in preparation for your two exams papers. Practice writing algorithms and using to answer questions.		
	➢ Exam skills and misconception • Practice exam papers (past & Exampro)	➢ Multiple Choice Question Activities ➢ Teacher/pupil questioning ➢ Exam style question practice (homework's and in class)	➢ Teacher demonstration and explanations
Summer 1	**Getting Ready for the exam** External exams begin		
	➢ Exam practise & skills	➢ Practise exam papers and questions ➢ Timed responses ➢ MCQ practise ➢ Marking activities ➢ Examiner's report	➢ Teacher demonstration and explanations
Summer 2	**Course complete**		