



Computer Science Curriculum Intent and Assessment Calendar 2026-27



ASSESSMENT WINDOW 1										ASSESSMENT WINDOW 2										ASSESSMENT WINDOW 3																			
Autumn Term										Spring Term										Summer Term																			
Block 1					Block 2					Block 3					Block 4					Block 5																			
Year 5																																							
Intro to Computers					PROGRESS CHECK NOV (Topic Test)	Networks					Vector Drawing					PROGRESS CHECK FEB (Topic Test)	Flat File Databases					PROGRESS CHECK JUNE (Topic Test)	Programming: Crumble Kits																
How to use the computing equipment safely and respectfully. Learning how to send an email, access school specific apps/websites and basic functions of the different Microsoft apps (Word, PowerPoint, Excel)						Explore digital devices, with an initial focus on inputs, processes and outputs. Pupils will be introduced to computer networks, including devices that make up the network.					Pupils will learn how to use different drawing tools to help them create images. Learners recognise that images in vector drawings are created using shapes and lines.						. Look at how a database can be used to organise data in records. Use tools to order and answer questions about data. Create graphs and charts from their data.						Pupils will explore physical computing through a Crumble controller. They are introduced to conditions and repetition as well as selection																
1	2	3	4	5		6	7				1	2	3	4	5		6	7					1	2	3	4	5	6	7				1	2	3	4	5	6	7
Year 6																																							
Networks					PROGRESS CHECK NOV (Topic Test)	Programming: Crumble Kits					Spreadsheets					PROGRESS CHECK FEB (Topic Test)	3D Modelling					PROGRESS CHECK JUNE (Topic Test)	Programming: MicroBits																
Pupils will explore how data is transferred over the internet. They will understand how the internet facilitates communication and collaboration and complete shared projects online.						Pupils will explore physical computing through a Crumble controller. They are introduced to conditions and repetition as well as selection					Introduction to Spreadsheets in Excel. Understanding how spreadsheets can be used. Formatting and entering specific formulas.						During this unit, pupils will develop their knowledge and understanding of using a computer to produce 3D models using online software TinkercAD.						Pupils will develop their understanding of physical coding, using Micro: Bits, to explore sequence, selection and iteration. They will explore the different inputs/outputs in a Micro:Bit and how they relate to everyday objects																
1	2	3	4	5		6	7				1	2	3	4	5		6	7					1	2	3	4	5	6	7				1	2	3	4	5	6	7
Year 7																																							
Programming Part 1					PROGRESS CHECK NOV (Topic Test)	Clear messages in Digital Media					Spreadsheets					PROGRESS CHECK FEB (Topic Test)	Networks					PROGRESS CHECK JUNE (Topic Test)	Programming Part 2																
Pupils will build on prior knowledge of sequence, variables and selection. They will understand and learn about count-controlled iteration and develop programming knowledge.						Pupils will develop a range of skills across different pieces of software. They will create posters and slides, plan and implement specific tasks and self-assess their work against success criteria					Pupils will build on prior knowledge of spreadsheets to collect, analyse and manipulate data, understanding how to convert data into graphs and charts.						Recognising networking hardware and explaining how networking components are used						How to design, use and evaluate computational abstraction when making an advanced game in Scratch programming.																
1	2	3	4	5		6	7	8	9		1	2	3	4	5		6	7	8	9			1	2	3	4	5	6	7	8	9		1	2	3	4	5	6	7
Year 8																																							
Media- vector graphics					PROGRESS CHECK NOV (Topic Test)	Developing for the web					Representations					PROGRESS CHECK FEB (Topic Test)	Mobile App Development					PROGRESS CHECK JUNE (Topic Test)	Intro to Python Programming																
Pupils will become familiar with digital graphic design specifically looking at making logos, illustrations and icons						Pupils will gain an understanding of the different technologies that make up the internet, the WWW, HTML and CSS. Pupils will develop a functioning website.					Representing numbers and text using binary digits- understanding how computers work and process data						Pupils will create their own mobile app, learning how to use Java Script and develop programming concepts.						Applying the programming constructs of sequence, selection, and iteration in Python.																
1	2	3	4	5		6	7	8	9		1	2	3	4	5		6	7	8	9			1	2	3	4	5	6	7	8	9		1	2	3	4	5	6	7

Computer Science programme of study: Key Stages 2 and 3

National curriculum in England

Purpose of study

Computer Science enables pupils to use computational thinking and creativity to understand and change the world. Computer Science has deep links with mathematics, science, and design and technology, and provides insights into both natural and artificial systems. Pupils are taught how digital systems work and how to program in different languages. Computer Science also ensures that pupils become digitally literate – able to use, and express themselves and develop their ideas through, information and communication technology – at a level suitable for the future workplace and as active participants in a digital world.

Aims

Over the four years of study at St Edwards, our curriculum will cover the main areas of Computer Science:

- **Algorithms** — Be able to comprehend, design, create and evaluate algorithms.
- **Computer networks** — Understand how networks can be used to retrieve and share information, and how they come with associated risks
- **Computer systems** — Understand what a computer is, and how its constituent parts function together as a whole
- **Creating media** — Select and create a range of media including text, images, sounds, and video
- **Data and information** — Understand how data is stored, organised, and used to represent real-world artefacts and scenarios
- **Design and development** — Understand the activities involved in planning, creating, and evaluating computing artefacts
- **Effective use of tools** — Use software tools to support computing work
- **Impact of technology** — Understand how individuals, systems, and society as a whole interact with computer systems
- **Programming** — Create software to allow computers to solve problems
- **Safety and security** — Understand risks when using technology

Spiral Delivery

The units for key stage 2 are based on a spiral curriculum. This means that each of the themes is revisited at least once in each year group. This style of curriculum design reduces the amount of knowledge lost through forgetting, as topics are revisited yearly. It also ensures that connections are made even if different teachers are teaching the units within a theme in consecutive years.

Inclusive and ambitious

The Curriculum has been resourced to support all pupils. Each lesson is sequenced so that it builds on the learning from the previous lesson, and where appropriate, activities are scaffolded so that all pupils can succeed and thrive. Scaffolded activities provide pupils with extra resources, such as visual prompts, to reach the same learning goals as the rest of the class. Exploratory tasks foster a deeper understanding of a concept, encouraging pupils to apply their learning in different contexts and make connections with other learning experiences.

Assessment Objectives

Key Stage 2

- A01 use technology safely, respectfully and responsibly; recognise acceptable/unacceptable behaviour; identify a range of ways to report concerns about content and contact
- A02 use search technologies effectively, appreciate how results are selected and ranked, and be discerning in evaluating digital content
- A03 select, use and combine a variety of software (including internet services) on a range of digital devices to design and create a range of programs, systems and content that accomplish given goals, including collecting, analysing, evaluating and presenting data and information
- A04 understand computer networks, including the internet; how they can provide multiple services, such as the World Wide Web, and the opportunities they offer for communication and collaboration
- A05 use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs
- A06 use sequence, selection, and repetition in programs; work with variables and various forms of input and output
- A07 design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts

Key Stage 3

- A01 understand a range of ways to use technology safely, respectfully, responsibly and securely, including protecting their online identity and privacy; recognise inappropriate content, contact and conduct, and know how to report concerns
- A02 understand the hardware and software components that make up computer systems, and how they communicate with one another and with other systems
- A03 understand simple Boolean logic [for example, AND, OR and NOT] and some of its uses in circuits and programming; understand how numbers can be represented in binary, and be able to carry out simple operations on binary numbers [for example, binary addition, and conversion between binary and decimal]
- A04 understand how instructions are stored and executed within a computer system; understand how data of various types (including text, sounds and pictures) can be represented and manipulated digitally, in the form of binary digits
- A05 create, reuse, revise and repurpose digital artefacts for a given audience, with attention to trustworthiness, design and usability
- A06 undertake creative projects that involve selecting, using, and combining multiple applications, preferably across a range of devices, to achieve challenging goals, including collecting and analysing data and meeting the needs of known users
- A07 design, use and evaluate computational abstractions that model the state and behaviour of real-world problems and physical systems
- A08 use 2 or more programming languages, at least one of which is textual, to solve a variety of computational problems; make appropriate use of data structures [for example, lists, tables or arrays]; design and develop modular programs that use procedures or functions
- A09 understand several key algorithms that reflect computational thinking [for example, ones for sorting and searching]; use logical reasoning to compare the utility of alternative algorithms for the same problem