Pope Francis
Catholic Multi Academy Trust

The Salesian Academy of
St. John
Bosco

Computing Curriculum Map

Vision Statement

At the Salesian Academy of St John Bosco, we are dedicated to fostering a deep appreciation and proficiency in computer science, programming, and media among all our students. We firmly believe that with dedication and effort, every student can excel in these fields. Through engaging teaching practices and a progressive curriculum, we aim to cultivate confident, enthusiastic, and inquisitive learners who are well-equipped for the digital age. Our curriculum is designed to facilitate skill progression year by year, ensuring that students build upon their knowledge and abilities over time in computer science, programming, and media studies.

We hold high aspirations for all students, providing inclusive learning experiences and appropriate challenges tailored to individual needs. Our goal is to empower students to become self-motivated, independent, and resilient problem solvers, proficient in foundational computational skills, programming languages, and media literacy, capable of adapting to new challenges. We challenge our students to excel as computer scientists and media creators, demonstrating fluency, accuracy, and confidence in applying their skills across diverse contexts.

Grounded in Salesian values, our approach to computer science and media education emphasizes love, care, empowerment through hard work, community, inclusivity, inspiration, aspirations, service, confidence, and competence, shaping students into ethical and innovative leaders in the digital age.

Year 7 Curriculum Intent

In Year 7, students will explore computer science, media literacy, and network basics, gaining practical skills through hands-on projects and problem-solving tasks. They will learn programming fundamentals, delve into digital media creation and consumption, and understand the fundamentals of computer networks. Through regular tasks and activities, they will reinforce their learning, developing a strong foundation in these areas. By the end of the year, students will be well-prepared to build upon their knowledge in the Year 8 curriculum and continue their exploration of the dynamic field of technology.

Year 7 Curriculum Map - Autumn			
Topic Details	**No. of Lessons**	**Rational/Sequencing**	**N.C. Coverage**
Clear messaging in digital media This unit is designed to build upon learners' experience in key stage 2. It requires learners to use a range of different skills across several pieces of software. Learners will work between different applications to create a poster and slides on a given theme. The unit is designed so that learners can concentrate on applying skills that they may have previously learnt as well as those learnt in the unit. Learners are given clear tasks for which they need to first plan and then implement a solution.	6	Choose search terms relating to a particular issue Use tools to copy an image into another application Identify key features of a good poster Plan a poster to clearly convey a message Choose and download a suitable image Create a poster using a desktop publishing application Modify a logo using a graphic editing program Choose how to combine text and graphics in a slide Use digital tools to provide feedback on design choices Plan a consistent layout for a set of slides Modify a logo so that it fits in with the planned slide styles Create a styled set of slides based on a plan Search for suitable text for slides Search for and add a suitable image Evaluate content against a rubric Plan how to deliver a presentation Explain your work to others through a presentation Evaluate your work against a rubric	● Create, reuse, revise, and repurpose digital artefacts for a given audience, with attention to trustworthiness, design, and usability ● 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

Year 7 Curriculum Map - Spring			
Topic Details	**No. of Lessons**	**Rational/Sequencing**	**N.C. Coverage**
Networks from Semaphores to the internet As networks have evolved, society has become increasingly reliant on the services that they provide. They have changed the way we learn, work, play, and communicate. This unit begins by defining a network and addressing the benefits of networking, before covering how data is transmitted across networks using protocols. The types of hardware required are explained, as is wired and wireless data transmission. Learners will develop an understanding of the terms 'internet' and 'World Wide Web', and of the key services and protocols used. Practical exercises are included throughout to help strengthen understanding.	6	Define what a computer network is and explain how data is transmitted between computers across networks Define 'protocol' and provide examples of non-networking protocols List examples of the hardware necessary for connecting devices to networks Compare wired to wireless connections and list examples of specific technologies currently used to implement such connections Define 'bandwidth', using the appropriate units for measuring the rate at which data is transmitted, and discuss familiar examples where bandwidth is important Define what the internet is Explain how data travels between computers across the internet Describe key words such as 'protocols', 'packets', and 'addressing' Explain the difference between the internet, its services, and the World Wide Web Describe how services are provided over the internet	● Understand the hardware and software components that make up computer systems, and how they communicate with one another and with other systems

		List some of these services and the context in which they are used Explain the term 'connectivity' as the capacity for connected devices ('Internet of Things') to collect and share information about me with or without my knowledge (including microphones, cameras, and geolocation) Describe how internet-connected devices can affect me Describe components (servers, browsers, pages, HTTP and HTTPS protocols, etc.) and how they work together	

Year 7 Curriculum Map - Summer			
Topic Details	**No. of Lessons**	**Rational/Sequencing**	**N.C. Coverage**
Programming in Scratch 1 This unit is the first programming unit of KS3. The aim of this unit and the following unit ('programming 2') is to build learners' confidence and knowledge of the key programming constructs. Importantly, this unit does not assume any previous programming experience, but it does offer learners	6	Compare how humans and computers understand instructions (understand and carry out) Define a sequence as instructions performed in order, with each executed in turn Predict the outcome of a simple sequence Modify a sequence Define a variable as a name that refers to data being stored by the computer	● Use two or more programming languages, at least one of which is textual, to solve a variety of computational problems; make appropriate use of data structures (e.g. lists, tables, or arrays); design and develop modular programs that use procedures or functions

the opportunity to expand on their knowledge throughout the unit. The main programming concepts covered in this unit are sequencing, variables, selection, and count-controlled iteration. All of the examples and activities for this unit use Scratch 3.		Recognise that computers follow the control flow of input/process/output Predict the outcome of a simple sequence that includes variables Trace the values of variables within a sequence Make a sequence that includes a variable Define a condition as an expression that will be evaluated as either true or false Identify that selection uses conditions to control the flow of a sequence Identify where selection statements can be used in a program Modify a program to include selection Create conditions that use comparison operators (>,<,=) Create conditions that use logic operators (and/or/not) Identify where selection statements can be used in a program that include comparison and logical operators Define iteration as a group of instructions that are repeatedly executed Describe the need for iteration Identify where count-controlled iteration can be used in a program Implement count-controlled iteration in a program Detect and correct errors in a program (debugging) Independently design and apply programming constructs to solve a problem (subroutine,	● Understand several key algorithms that reflect computational thinking; use logical reasoning to compare the utility of alternative algorithms for the same problem ● Understand simple Boolean logic (e.g. and, or, and not) ● Create, reuse, revise, and repurpose digital artefacts for a given audience, with attention to trustworthiness, design, and usability

		selection, count-controlled iteration, operators, and variables)	

Year 8 Curriculum Intent

In Year 8, students will further their exploration of technology, focusing on media literacy, spreadsheet skills, programming, and web development. Building on their foundational knowledge from Year 7, students will delve deeper into digital media creation and analysis, mastering the tools and techniques essential for effective communication in the digital age. They will also learn to utilize spreadsheets for data management and analysis, developing essential computational and analytical skills. Through hands-on programming projects, students will expand their understanding of algorithms, logic, and problem-solving, preparing them for more advanced studies in computer science. Additionally, students will explore the fundamentals of web development, learning to design and create interactive websites using HTML and CSS. By engaging in a diverse range of projects and activities, students will cultivate their creativity, critical thinking, and technical skills, paving the way for continued growth and exploration in Year 9 and beyond.

Year 8 Curriculum Map - Autumn

Topic Details	No. of Lessons	Rational/Sequencing	N.C. Coverage
Using Media During this unit, learners develop their understanding of information technology and digital literacy skills. They will use the skills learnt across the unit to create a blog post about a real-world cause that they would like to gain support for. Learners will develop software formatting skills and explore concerns surrounding the use of other	6	Select the most appropriate software to use to complete a task Identify the key features of a word processor Apply the key features of a word processor to format a document Evaluate formatting techniques to understand why we format documents Select appropriate images for a given context Apply appropriate formatting techniques	• 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 • Create, reuse, revise, and repurpose digital artefacts for a given audience, with attention to

people's work, including licensing and legal issues.		Demonstrate an understanding of licensing issues involving online content by applying appropriate Creative Commons licences Demonstrate the ability to credit the original source of an image Critique digital content for credibility Apply techniques to identify whether or not a source is credible Apply referencing techniques and recognise the concept of plagiarism Evaluate online sources for use in own work Construct a blog using appropriate software Create content for a blog based on credible sources Apply referencing techniques that credit authors appropriately Design the layout of the content to make it suitable for the audience Construct a blog using appropriate software Create content for a blog based on credible sources Apply referencing techniques that credit authors appropriately Design the layout of the content to make it suitable for the audience	trustworthiness, design, and usability
Modelling Data with Spreadsheets	6	Identify columns, rows, cells, and cell references in spreadsheet software Use formatting techniques in a spreadsheet	● Design, use, and evaluate computational abstractions that

The spreadsheet unit for Year 7 takes learners from having very little knowledge of spreadsheets to being able to confidently model data with a spreadsheet. The unit uses engaging activities to progress learners from using basic formulas to writing their own COUNTIF statements. This unit will give learners a good set of skills that they can use in computing lessons and in other subject areas.		Use basic formulas with cell references to perform calculations in a spreadsheet (+, -, *, /) Use the autofill tool to replicate cell data Explain the difference between data and information Explain the difference between primary and secondary sources of data Collect data Analyse data Create appropriate charts in a spreadsheet Use the functions SUM, COUNTA, MAX, and MIN in a spreadsheet Analyse data Use a spreadsheet to sort and filter data Use the functions AVERAGE, COUNTIF, and IF in a spreadsheet Use conditional formatting in a spreadsheet Apply all of the spreadsheet skills covered in this unit	model the state and behaviour of real-world problems and physical systems ● 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

Year 8 Curriculum Map - Spring			
Topic Details	**No. of Lessons**	**Rational/Sequencing**	**N.C. Coverage**
Programming in Scratch 2	6	Define a subroutine as a group of instructions that will run when called by the main program or other subroutines	● To use two or more programming languages, at least one of which is textual, to solve a variety of

This unit begins right where 'Programming 1' left off. Learners will build on their understanding of the control structures' sequence, selection, and iteration (the big three), and develop their problem-solving skills. Learners will learn how to create their own subroutines, develop their understanding of decomposition, learn how to create and use lists, and build upon their problem-solving skills by working through a larger project at the end of the unit.		Define decomposition as breaking a problem down into smaller, more manageable subproblems Identify how subroutines can be used for decomposition Identify where condition-controlled iteration can be used in a program Implement condition-controlled iteration in a program Evaluate which type of iteration is required in a program Define a list as a collection of related elements that are referred to by a single name Describe the need for lists Identify when lists can be used in a program Use a list Decompose a larger problem into smaller sub-problems Apply appropriate constructs to solve a problem	computational problems; to make appropriate use of data structures (for example, lists, tables, or arrays); to design and develop modular programs that use procedures or functions ● To understand several key algorithms that reflect computational thinking; use logical reasoning to compare the utility of alternative algorithms for the same problem ● To understand simple Boolean logic (for example, AND, OR, and NOT) ● To create, reuse, revise, and repurpose digital artefacts for a given audience, with attention to trustworthiness, design, and usability
Developing for the Web In this unit, learners will explore the technologies that make up the internet and World Wide Web. Starting with an exploration of the building blocks of the World Wide Web, HTML, and CSS, learners will investigate how websites are catalogued and organised for	6	Describe what HTML is Use HTML to structure static web pages Modify HTML tags using inline styling to improve the appearance of web pages Display images within a web page Apply HTML tags to construct a web page structure from a provided design	● Create, reuse, revise, and repurpose digital artefacts for a given audience, with attention to trustworthiness, design, and usability.

effective retrieval using search engines. By the end of the unit, learners will have a functioning website.		Describe what CSS is Use CSS to style static web pages Assess the benefits of using CSS to style pages instead of in-line formatting Describe what a search engine is Explain how search engines 'crawl' through the World Wide Web and how they select and rank results Analyse how search engines select and rank results when searches are made Use search technologies effectively Discuss the impact of search technologies and the issues that arise by the way they function and the way they are used Create hyperlinks to allow users to navigate between multiple web pages Implement navigation to complete a functioning website	

Year 8 Curriculum Map - Summer			
Topic Details	**No. of Lessons**	**Rational/Sequencing**	**N.C. Coverage**
Representations This unit conveys essential knowledge relating to binary representations. The activities gradually introduce learners	6	List examples of representations Recall that representations are used to store, communicate, and process information Provide examples of how different representations are appropriate for different tasks	• Understand how data of various types (including text, sounds and pictures) can be represented and manipulated digitally, in the form of binary digits

to binary digits and how they can be used to represent text and numbers. The concepts are linked to practical applications and problems that the learners are familiar with.		Recall that characters can be represented as sequences of symbols and list examples of character coding schemes Measure the length of a representation as the number of symbols that it contains Provide examples of how symbols are carried on physical media Explain what binary digits (bits) are, in terms of familiar symbols such as digits or letters Measure the size or length of a sequence of bits as the number of binary digits that it contains Describe how natural numbers are represented as sequences of binary digits Convert a decimal number to binary and vice versa Convert between different units and multiples of representation size Provide examples of the different ways that binary digits are physically represented in digital devices	
Mobile App Development In a world where there's an app for every possible need, this unit aims to take the learners from designer to project manager to developer in order to create their own mobile app. Using App Lab from code.org, learners will familiarise themselves with the coding environment and have an opportunity to build on the programming concepts	6	Identify when a problem needs to be broken down Implement and customise GUI elements to meet the needs of the user Recognise that events can control the flow of a program Use user input in an event-driven programming environment Use variables in an event-driven programming environment	● Design, use, and evaluate computational abstractions that model the state and behaviour of real-world problems and physical systems ● Use two or more programming languages, at least one of which is textual, to solve a variety of computational problems; make

they used in previous units before undertaking their project. Learners will work in pairs to consider the needs of the user; decompose the project into smaller, more manageable parts; use the pair programming approach to develop their app together; and finish off by evaluating the success of the project against the needs of the user.		Develop a partially complete application to include additional functionality Identify and fix common coding errors Pass the value of a variable into an object Establish user needs when completing a creative project Apply decomposition to break down a large problem into more manageable steps Use user input in a block-based programming language Use a block-based programming language to create a sequence Use variables in a block-based programming language Use a block-based programming language to include sequencing and selection Use user input in a block-based programming language Use variables in a block-based programming language Reflect and react to user feedback Use a block-based programming language to include sequencing and selection Use user input in a block-based programming language Use variables in a block-based programming language Evaluate the success of the programming project	appropriate use of data structures [for example, lists, tables, or arrays]; design and develop modular programs that use procedures or functions ● Understand several key algorithms that reflect computational thinking; use logical reasoning to compare the utility of alternative algorithms for the same problem ● Create, reuse, revise, and repurpose digital artefacts for a given audience, with attention to trustworthiness, design, and usability

Year 9 Curriculum Intent

In Year 9, students will deepen their exploration of technology, with a focus on vector graphics, programming in Python, animations, and cybersecurity. Building on their prior knowledge from Year 8, students will delve into the world of vector graphics, mastering techniques for creating and manipulating scalable images for various applications. They will also advance their programming skills by learning Python, a versatile and widely-used language in the field of computer science. Through hands-on projects, students will explore the fundamentals of animation, gaining insight into techniques for creating dynamic and interactive visual content. Additionally, students will study cybersecurity, learning about the importance of protecting digital assets and the principles of secure computing practices. By engaging in challenging projects and practical exercises, students will develop a deeper understanding of technology's role in society, preparing any pupils who choose to study a media or computing qualification in Key Stage 4.

Year 9 Curriculum Map - Autumn			
Topic Details	**No. of Lessons**	**Rational/Sequencing**	**N.C. Coverage**
Vector Graphics This unit offers learners the opportunity to design graphics using vector graphic editing software. By the end of the unit learners will have produced an illustration, a logo, or some icons using vector graphics. Vector graphics can be used to design anything from logos and icons to posters, board games, and complex illustrations. Through this unit, learners will be able to better understand the processes involved in creating such graphics and will be provided with the knowledge and tools to create their own.	6	Use tools to draw and modify shapes Change the position and rotation shapes Explain how z-order determines what is visible Use tools to align and distribute objects to create uniformity Explain how grouping can be used to work with several objects at once Combine two shapes using union, intersection, and difference Explain that vector graphics are made up of paths Create and modify straight and curved paths Change shapes to paths and edit them Choose a project and plan a design Combine tools and techniques to create a vector image Evaluate the project against its given purpose Explain how markup defines what a vector graphic looks like Change an object by modifying its markup Plan improvements and implement them to develop a project Explain key differences between vector and bitmap images Outline which image type best suits which uses Evaluate their image against a rubric	• 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 • Create, reuse, revise, and repurpose digital artefacts for a given audience, with attention to trustworthiness, design, and usability

Introduction to Python This unit introduces learners to text-based programming with Python. The lessons form a journey that starts with simple programs involving input and output, and gradually moves on through arithmetic operations, randomness, selection, and iteration. Emphasis is placed on tackling common misconceptions and elucidating the mechanics of program execution. A range of pedagogical tools is employed throughout the unit, with the most prominent being pair programming, live coding, and worked examples	6	Describe what algorithms and programs are and how they differ Recall that a program written in a programming language needs to be translated in order to be executed by a machine Write simple Python programs that display messages, assign values to variables, and receive keyboard input Locate and correct common syntax errors Describe the semantics of assignment statements Use simple arithmetic expressions in assignment statements to calculate values Receive input from the keyboard and convert it to a numerical value Use relational operators to form logical expressions Use binary selection (if, else statements) to control the flow of program execution Generate and use random integers Use multi-branch selection (if, elif, else statements) to control the flow of program execution Describe how iteration (while statements) controls the flow of program execution Use iteration (while loops) to control the flow of program execution Use variables as counters in iterative programs	● Can understand and apply the fundamental principles and concepts of computer science, including abstraction, logic, algorithms and data representation ● Can analyse problems in computational terms, and have repeated practical experience of writing computer programs in order to solve such problems ● Use two or more programming languages, at least one of which is textual, to solve a variety of computational problems ● Understand several key algorithms that reflect computational thinking; use logical reasoning to compare the utility of alternative algorithms for the same problem ● Understand how instructions are stored and executed within a computer system ● Design, use and evaluate computational abstractions that model the state and behaviour of real-world problems and physical systems

		Combine iteration and selection to control the flow of program execution Use Boolean variables as flags	

Year 9 Curriculum Map - Spring			
Topic Details	**No. of Lessons**	**Rational/Sequencing**	**N.C. Coverage**
Animations Films, television, computer games, advertising, and architecture have been revolutionised by computer-based 3D modelling and animation. In this unit learners will discover how professionals create 3D animations using the industry-standard software package, Blender. By completing this unit learners will gain a greater understanding of how this important creative field is used to make the media products that we consume. Sessions will take learners through the basics of modelling, texturing, and animating; outputs will include 3D models, short videos, and VR. Links are made throughout to computer science, computational thinking, and the world of work. Tools and techniques learnt in this unit can also be used for 3D printing.	6	Add, delete, and move objects Scale and rotate objects Use a material to add colour to objects Add, move, and delete keyframes to make basic animations Play, pause, and move through the animation using the timeline Create useful names for objects Join multiple objects together using parenting Use edit mode and extrude Use loop cut and face editing Apply different colours to different parts of the same model Use proportional editing Use the knife tool Use subdivision Add and edit set lighting Set up the camera Compare different render modes Create a 3–10 second animation Render out the animation	● Create, reuse, revise, and repurpose digital artefacts for a given audience, with attention to trustworthiness, design, and usability

Python: Sequences of Data This unit introduces learners to how data can be represented and processed in sequences, such as lists and strings. The lessons cover a spectrum of operations on sequences of data, that range from accessing an individual element to manipulating the entire sequence. Great care has been taken so that the selection of problems used in the programming tasks are realistic and engaging: learners will process solar system planets, book texts, capital cities, leaked passwords, word dictionaries, ECG data, and more. The Year 7 and 8 Programming units are prerequisites for this unit. It is assumed that learners are already able to write Python programs that display messages, receive keyboard input, use simple arithmetic expressions, and control the flow of program execution through selection and iteration structures.	6	Write programs that display messages, receive keyboard input, and use simple arithmetic expressions in assignment statements Use selection (if-elif-else statements) to control the flow of program execution Locate and correct common syntax errors Create lists and access individual list items Perform common operations on lists or individual items Use iteration (while statements) to control the flow of program execution Perform common operations on lists or individual items Perform common operations on strings or individual characters Use iteration (for statements) to iterate over list items Perform common operations on lists or strings Use iteration (for loops) to iterate over lists and strings Use variables to keep track of counts and sums Combine key programming language features to develop solutions to meaningful problems	● Can understand and apply the fundamental principles and concepts of computer science, including abstraction, logic, algorithms, and data representation ● Can analyse problems in computational terms, and have repeated practical experience of writing computer programs in order to solve such problems ● Use two or more programming languages, at least one of which is textual, to solve a variety of computational problems ● Understand how instructions are stored and executed within a computer system ● Understand several key algorithms that reflect computational thinking; use logical reasoning to compare the utility of alternative algorithms for the same problem ● Design, use, and evaluate computational abstractions that model the state and behaviour of real-world problems and physical systems

Year 9 Curriculum Map - Summer			
Topic Details	**No. of Lessons**	**Rational/Sequencing**	**N.C. Coverage**
Representations: Audiovisual In this unit, learners will focus on digital media such as images and sounds, and discover the binary digits that lie beneath these types of media. Just like in the previous unit, where learners examined characters and numbers, the ideas that learners need to understand are not really new to them. You will draw on familiar examples of composing images out of individual elements, mixing elementary colours to produce new ones, and taking samples of analogue signals, to illustrate these ideas and bring them together in a coherent narrative. This unit also has a significant practical aspect. Learners will use relevant software (GIMP and Audacity, in this case) to manipulate images and sounds and get an idea of how the underlying principles of digital representations are applied in real settings	6	Describe how digital images are composed of individual elements Recall that the colour of each picture element is represented using a sequence of binary digits Define key terms such as 'pixels', 'resolution', and 'colour depth' Describe how an image can be represented as a sequence of bits Describe how colour can be represented as a mixture of red, green, and blue, with a sequence of bits representing each colour's intensity Compute the representation size of a digital image, by multiplying resolution (number of pixels) with colour depth (number of bits used to represent the colour of individual pixels) Describe the trade-off between representation size and perceived quality for digital images Perform basic image editing tasks using appropriate software and combine them in order to solve more complex problems requiring image manipulation Explain how the manipulation of digital images amounts to arithmetic operations on their digital representation Describe and assess the creative benefits and ethical drawbacks of digital manipulation (Education for a Connected World)	● Understand how data of various types (including text, sounds and pictures) can be represented and manipulated digitally, in the form of binary digits

Cybersecurity This unit takes the learners on an eye-opening journey of discovery about techniques used by cybercriminals to steal data, disrupt systems, and infiltrate networks. The learners will start by considering the value of their data to organisations and what they might use it for. They will then look at social engineering techniques used by cybercriminals to try to trick users into giving away their personal data. The unit will look at the more common cybercrimes such as hacking, DDoS attacks, and malware, as well as looking at methods to protect ourselves and our networks against these attacks.	6	Explain the difference between data and information Critique online services in relation to data privacy Identify what happens to data entered online Explain the need for the Data Protection Act Recognise how human errors pose security risks to data Implement strategies to minimise the risk of data being compromised through human error Define hacking in the context of cyber security Explain how a DDoS attack can impact users of online services Identify strategies to reduce the chance of a brute force attack being successful Explain the need for the Computer Misuse Act List the common malware threats Examine how different types of malware causes problems for computer systems Question how malicious bots can have an impact on societal issues Compare security threats against probability and the potential impact to organisations Explain how networks can be protected from common security threats Identify the most effective methods to prevent cyberattacks	● 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