ST JOSEPH
Catholic Multi Academy Trust

SPERABO IN EUM

Computing, Computer Science

Long-Term Plan

Signatory	Full Name
Head of Department	Richard Headley
SLT for Curriculum	Andrew Doyle
Headteacher	Giselle Lynch

ST JOSEPH
Catholic Multi Academy Trust

Table of Contents

Catholic Social Teachings (CST)

ST JOSEPH
Catholic Multi Academy Trust

Computing, Computer Science and Business Intent

Vision and Values
(What is your subject? What do you teach? And why do you teach it? What are your ambitions for your pupils? What makes your subject, your subject? What do students in your subject think like?) Computing & Computer Science is an ever-evolving subject at the forefront of modern life. The ambitions are for Computer Science & Computing to be a strongly chosen subject at Key Stage 4.

Curriculum Design

Design
(How is your Curriculum designed? How is it sequenced? How is are key concepts developed? How does your curriculum design support long-term memory retention? How does your curriculum design make use of inter-leaved practice or spaced practice?) **KS3** "Building/Interleaving Curriculum" based on the different strands of Computing: Networks and the Internet, Computer Systems, Collecting and Presenting Data, Data Representation, Algorithms and Programming Techniques and Creating Programs. These strands all link directly to the KS3 National Curriculum ensuring a broad and in-depth coverage of required content throughout the 3-year KS3. These strands also have links to the curriculum specification design at KS4 in GCSE Computer Science. Each strand is covered throughout learners KS3 seeing skills revisited and built upon each year. In addition to revisiting strands throughout the curriculum design, skills are also revisited from across the topic areas each lesson through knowledge retrieval tasks at the beginning of each lesson to embed and support long-term memory retention.

ST JOSEPH
Catholic Multi Academy Trust

KS3 Computing Key Concepts

1. **Networks and the Internet**
   An introduction to networks and the internet including how they work and impact on our day to day lives when using computing equipment both at school, at home and in the wider world.
2. **Computer Systems**
   Students will understand what a computer is, and how its constituent parts function together as a whole.
3. **Collecting & Presenting Data**
   Students will understand how to collect data from different sources and present data using different software packages, mainly using the Microsoft Office suite of programs.
4. **Data Representation**
   Students will understand how data is stored, organised, and used to represent real-world artefacts and scenarios. Data representation in the form of Binary is used throughout the Key stage 3 course.
5. **Algorithms**
   Students will be able to comprehend, design, create, and evaluate algorithms.
6. **Programming Techniques and Creating Programs**
   Students will create software to allow computers to solve problems using a range of programming languages and software from Scratch to Python

GCSE Computer Science Key Concepts

Computer Systems

1. **Systems Architecture**
   Architecture of the CPU, CPU Performance, Embedded Systems.
2. **Memory and Storage**
   Primary Storage (memory), Secondar Storage, Units, Data Storage, Compression.
3. **Computer Networks, Connections and Protocols**
   Networks & topologies, Wired & wireless network protocols & layers.
4. **Network Security**
   Threats to computer systems & networks, identifying & preventing vulnerabilities.
5. **Systems Software**
   Operating systems, utility software.
6. **Ethical, Legal, Cultural and Environmental Impacts of Digital Technology**
   Ethical, legal, cultural & environmental impacts of technology.

Computational Thinking, Algorithms and Programming

1. **Algorithms**
   Computational thinking, designing, creating & refining algorithms, Searching & storing algorithms.
2. **Programming Fundamentals**
   Programming fundamentals, Data types, Additional programming techniques.
3. **Producing Robust Programs**
   Defensive design, testing.
4. **Boolean Logic**
   Boolean logic.
5. **Programming Languages, and Integrated Development Environments**
   Languages, Integrated Development Environment (IDE).

ST JOSEPH
Catholic Multi Academy Trust

Computing, Computer Science LTP

	Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer 2
Year 7	**Introduction to the St Augustine's school network** Log in, complex passwords, file names & folders **Collecting and Presenting Data** Spreadsheets	**Networks and the Internet** eSafety Social Media Online Privacy Laws	**Data Representation** Binary Numbers Characters	**Computer Systems** Data Inside a Computer Input Devices Output Devices	**Programming Techniques and Creating Programs** [Scratch] Data Types Operators Variables Inputs & Outputs Sequence Selection Iteration	
Year 8	**Collecting and Presenting Data** Target Audience Collecting Data Analysing Data Presenting Data	**Collecting and Presenting Data** Web Design HTML	**Networks and the Internet** Types of Network Wired & Wireless Networks Network Topologies	**Computer Systems** The CPU Memory Internal Storage External Storage	**Algorithms** Decomposition Abstraction Flowcharts Pseudocode Linear Search Binary Search Bubble Sort Merge Sort	**Programming Techniques and Creating Programs** [Python] Data Types Operators Variables Inputs & Outputs Sequence Selection Iteration
Year 9	**Enterprise** Enterprises & Entrepreneur Finance Marketing Business plans & presentations	**Programming Techniques and Creating Programs** [Python] Data Types Operators Variables Inputs & Outputs Sequence Selection & Iteration	**Data Representation** Sound Compression Logic	**Collection & Presenting Data** Databases	**Networks and the Internet** The Internet Network Security Threats Network Security Measures Censorship & Surveillance	**Computer Systems** Software The Operating Systems

Year 10 **Computer Science**	2.4.1 Boolean Logic 1.2.3 Units 1.2.4 Data Storage (Numbers) 1.2.4 Data Storage (Characters) 1.2.4 Data Storage (Images) 1.2.4 Data Storage (Sound) 1.2.5 Compression 2.1.1 Computational Thinking 1.4.1 Threats to Computer Systems and Networks	1.1.1 Architecture of the CPU 1.1.2 CPU Performance 1.1.3 Embedded Systems 1.2.1 Primary Storage (Memory) 1.2.2 Secondary Storage 2.1.2 Designing, creating and refining algorithms	1.3.1 Networks and topologies 1.3.2 Wired and wireless networks, protocols and layers 2.1.2 Designing, creating and refining algorithms	1.4.1 Threats to computer systems and networks 1.4.2 Identifying and preventing vulnerabilities 2.1.2 Designing, creating and refining algorithms	1.5.1 Operating systems 1.5.2 Utility software 2.1.3 Searching and sorting algorithms	1.6.1 Ethical, legal, cultural and environmental impact 2.1.3 Searching and sorting algorithms
Year 11 **Computer Science**	2.5.1 Languages 2.5.2 The Integrated Development Environment (IDE) 2.2.1 Programming fundamentals 2.2.2 Data types	2.2.1 Programming fundamentals 2.2.2 Data types 2.2.3 Additional programming techniques	2.3.1 Defensive design 2.3.2 Testing	2.1.2 Designing, creating and refining algorithms 2.2.1 Programming fundamentals Revision	Revision	N/A

ST JOSEPH
Catholic Multi Academy Trust

GCSE Computer Science

		Year 1			Year 2		
		Lesson 1	Lesson 2	Lesson 3	Lesson 1	Lesson 2	Lesson 3
		Autumn Term					
04/09/2023	Week 1		Introduction/Admin	2.4.1 Logic Diagrams and Truth Tables	2.5.1 High and Low Level Languages	2.5.1 Translators	P1 - Component 2 Revision
11/09/2023	Week 2	2.4.1 Logic Circuits	2.4.1 Solving Problems		2.5.2 IDE's	2.2.1 Python	
18/09/2023	Week 3	2.4.1 Examination Masterclass	1.2.3 Units	1.2.4 Binary Conversion	2.2.1 Python	2.2.1 Python	P1 - Component 2 Revision
25/09/2023	Week 4	1.2.4 Binary Addition	1.2.4 Binary Shifts		2.2.1 Python	2.2.1 Python	
02/10/2023	Week 5	1.2.4 Hexadeimal	1.2.3 & 1.2.4 [N] Examination Masterclass	2.1.1 Computational Thinking	2.2.1 Python	2.2.1 Python	P1 - Component 1 Revision
09/10/2023	Week 6	1.2.4 Character Sets	1.2.4 Images		2.2.1 Python	2.2.1 Python Assessment	
16/10/2023	Week 7	1.2.4 Sound	1.2.5 Compression	2.1.2 Inputs > Processess > Outputs	Revision	Revision	P1 - Component 3 Revision
23/10/2023	Week 8	1.2.4 [C,I,S&C] Examination Masterclass	2.4.1, 1.2.3, 1.2.4 & 1.2.5 Assessment		Mock Exam	Mock Exams/Programming Challenges	
		Half Term					
06/11/2023	Week 9	1.1.1 Purpose of the CPU	1.1.1 CPU Components	2.1.2 Structure Diagrams	Mock Exams/Programming Challenges	Mock Exam Review	P1 - Component 4 Revision
13/11/2023	Week 10	1.1.1 Von-Neumann Architecture	1.1.2 CPU Performance		2.2.1 Python	2.2.1 Python	
20/11/2023	Week 11	1.1.3 Embedded Systems	1.1 .1, 1.1.2 & 1.1.3 Examination Masterclass	2.1.2 Flowcharts	2.2.1 Python	2.2.1 Python	P1 - Component 5 Revision
27/11/2023	Week 12	1.1.1, 1.1.2 & 1.1.3 Assessment	1.2.1 Primary Storage		2.2.1 Python	2.2.1 Python	
04/12/2023	Week 13	1.2.2 Secondary Storage	1.2.2 Secondary Storage	2.1.2 Pseudocode	2.2.1 Python	2.2.1 Python	P1 - Component 6 Revision
11/12/2023	Week 14	1.2.1 & 1.2.2 Examination Masterclass	Revision		2.2.1 Python	2.2.1 Python Assessment	
18/12/2023	Week 15	Mock Exam	Mock Exam Review	Algorithm Practice	Python Challenges	Python Challenges	P2 - Component 4 Revision
		End of Term					
		Spring Term					
08/01/2024	Week 1	1.3.1 Types of Networks	1.3.1 Performance Factors and Hardware		2.3.1 Defensive Design	2.3.1 Input Validation	
15/01/2024	Week 2	1.3.1 Hardware and Client-Server/Peer to Peer	1.3.1 The Internet	Algorithm Practice	2.3.1 Maintainability	2.3.2 The Purpose of Testing	P2 - Component 1 Revision
22/01/2024	Week 3	1.3.1 Network Topologies	1.3.2 Wired and Wireless Connections		2.3.2 Test Data	2.3.2 Refining Algorithms	
29/01/2024	Week 4	1.3.2 IP and MAC Addresses	1.3.2 Protocols and Layers	Algorithm Practice	2.3.1 & 2.3.2 Examination Masterclass	Revision	P2 - Component 5 Revision
05/02/2024	Week 5	1.3.2 Encryption	1.3.1 & 1.3.2 Examination Masterclass		Mock Exam	Mock Exam	
		Half Term					
19/02/2024	Week 6	1.3.1 & 1.3.2 Assessment	Review of Learning & Algorithm Practice	Algorithm Practice	Robust Programming Challenges	Robust Programming Challenges	Robust Programming Challenges
26/02/2024	Week 7	1.4.1 Forms of Attack	1.4.1 Social Engineering		Robust Programming Challenges	Robust Programming Challenges	
04/03/2024	Week 8	1.4.1 Brute Force Attacks	1.4.1 Denial of Service attacks	2.1.2 Trace Tables	Mock Exam Review	Mock Exam Review	Paper 2 Revision
11/03/2024	Week 9	1.4.1 Data Interception and Theft	1.4.1 SQL Injection		Paper 1 Revision	Paper 2 Revision	
18/03/2024	Week 10	1.4.2 Software Prevention Methods	1.4.2 Prevention Against Hacking	1.4.1 & 1.4.2 Examination Masterclass	Paper 1 Revision	Paper 2 Revision	Paper 2 Revision
25/03/2024	Week 11	1.4.1 & 1.4.2 Assessment	Review of Learning & Trace Table Practice		Paper 1 Revision	Paper 2 Revision	
		End of Term					
		Summer Term					
15/04/2024	Week 1	Revision	Revision	Revision	Paper 1 Revision	Paper 2 Revision	Paper 1 Revision
22/04/2024	Week 2	Revision	Revision		Paper 1 Revision	Paper 2 Revision	
29/04/2024	Week 3	Mock Exam	Mock Exams/Algorithms	2.1.3 Linear & Binary Searches	Paper 1 Revision	Paper 2 Revision	Paper 1 Revision
06/05/2024	Week 4	Mock Exams/Algorithms	Mock Exam Review		Paper 1 Revision	Paper 2 Revision	
13/05/2024	Week 5	1.5.1 Purpose of OS	1.5.1 Interfaces	2.1.3 Bubble/Merge/Insertion Sort	J277 COMPUTER SYSTEMS [PAPER 1]		
20/05/2024	Week 6	1.5.1 Memory, Multitasking & Peripherals	1.5.2 Utility Software		J277 COMPUTATIONAL THINKING, ALGORITHMS AND PROGRAMMING [PAPER 2]		
		Half Term					
03/06/2024	Week 7	1.5.2 Encryption, Defrag and Compression	1.5.1 & 1.5.2 Examination Masterclass	2.1.3 Bubble/Merge/Insertion Sort			
10/06/2024	Week 8	1.5.1 & 1.5.2 Assessment					
17/06/2024	Week 9	1.6.1 Introduction & Stakeholders	1.6.1 Ethical Issues	2.1.3 Bubble/Merge/Insertion Sort			
24/06/2024	Week 10	1.6.1 Society	1.6.1 Cultural Issues			N/A	
01/07/2024	Week 11	1.6.1 Environmental Issues	1.6.1 Privacy Issues	2.1.3 Bubble/Merge/Insertion Sort			
08/07/2024	Week 12	1.6.1 Legislation	1.6.1 Software Licenses				
15/07/2024	Week 13	1.6.1 Examination Masterclass	1.6.1 Examination Masterclass	2.1.3 Examination Masterclass			
		End of Term					

ST JOSEPH
Catholic Multi Academy Trust

KS3 Overview

What prior knowledge do students arrive with from KS2 and how is that knowledge built upon?

Due to the nature of Computing being a specialist subject area that historically primary and early years teaching staff have struggled with, the range of skills, knowledge and ability differs greatly between cohorts received from different feeder primary schools. Generally, students arrive with basic knowledge of "Information Technology" but lack on the Computer Science element of skills. With this in mind, each of the Computer Science related strands within our curriculum start from a foundation level, containing lots of stretch, challenge and extension activities, to cater for all academic diets. At KS2 students are expected to design & write programs, use sequencing and debugging, understand networks, search terminologies, use a variety of software & use technology safely & responsibly. We take all of these concepts from the KS2 national curriculum and build on these within our KS3 computing curriculum.

What is the rationale for the sequence of your three-year KS3 curriculum?

To aid consolidation and long-term memory/retrieval throughout students' academic years and beyond.

How is your curriculum assessed?

End of unit assessments which then feed into assessment grids where pupils are awarded with T-, T, or T+ based on their level of performance towards their end of KS4 minimum expected grade.

By the end of KS3 what should all students know and be able to do? And how do you ensure this is retained in the long-term memory of students?

The core of computing is computer science, in which pupils are taught the principles of information and computation, how digital systems work, and how to put this knowledge to use through programming. Building on this knowledge and understanding, pupils are equipped to use information technology to create programs, systems and a range of content. Computing 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.

How does your KS3 curriculum prepare students to be successful at KS4?

The KS3 curriculum covers the different strands of Computing: Networks and the Internet, Computer Systems, Collecting and Presenting Data, Data Representation, Algorithms and Programming Techniques and Creating Programs. These strands all link directly to the KS3 National Curriculum ensuring a broad and in-depth coverage of required content throughout the 3-year KS3. These strands also have links to the curriculum specification design at KS4 in GCSE Computer Science meaning that should they choose to take this as an option path they will have a strong foundation to enable them to excel in their learning. In addition to GCSE Computer Science pupils also have the option path of BTEC Enterprise. Whilst not required by the KS3 National Curriculum, our curriculum enables learners to have a flavour of the key concepts studied at KS4 to enable them to make an informed choice about the subject.

Take one key concept or discrete piece of substantive knowledge and explain how it progresses across the three years of your curriculum.

Computer programming develops through all 3 years at KS3 before developing further at KS4 if the student selects Computer Science as a subject at GCSE. In Year 7 students are taught Scratch and the basic principles linking to the Scratch programming software of data types, operators, variables, inputs / outputs, sequence, selection & iteration are taught. As students move into Year 8, they are introduced to Python as a programming language and algorithms. In Year 9, Python is used again with the subject areas of data types, operators, variables, inputs / outputs, sequence, selection & iteration are taught.

What cultural capital do students acquire from learning your curriculum?

Learning computer science provides students with various forms of cultural capital, including:

Technological Literacy
Understanding computer science provides students with the knowledge and skills to navigate and contribute to the increasingly digital world, enhancing their technological literacy.

Problem-Solving Skills
Computer science develops critical thinking and problem-solving abilities, which are applicable cross-curricular, helping students approach challenges with creativity and logic.

Coding Culture
Engaging with coding develops a culture of collaboration, innovation, and experimentation, encouraging students to explore new ideas.

Digital Citizenship
Computer science emphasises responsible and ethical use of technology, instilling values of digital citizenship such as online safety, privacy protection, and respect for intellectual property.

Cultural Understanding
Through coding and technology projects, students may explore cultural themes, histories, and perspectives, fostering a deeper understanding of diverse cultures and identities.

Year 7 Computing LTP

	Autumn 1		Autumn 2	Spring 1	Spring 2	Summer 1 & 2
Title	Introduction to the St Augustine's school network	Collecting and Presenting Data	Networks and the Internet	Data Representation	Computer Systems Data	Programming Techniques and Creating Programs
Substantive Knowledge Taught	Log in, complex passwords, file names & folders	Spreadsheets	eSafety Social Media Online Privacy Laws	Binary Numbers Characters	Inside a Computer Input Devices Output Devices	Scratch Data Types, Operators, Variables, Inputs and Outputs, Sequence, Selection, Iteration
Disciplinary Knowledge (inc. Key Concepts)	Spreadsheet formula Spreadsheet functions		Online privacy Computer related laws	Performing binary conversion Performing binary addition	The different input & output devices	Programming syntax Programming errors
Tier 3 Vocabulary	Cell Cell Reference Function Formula		Network Personal data Privacy Cyber IP Address	Binary Denary ASCII	Input Output Peripheral	Operators, Variables, Inputs and Outputs, Sequence, Selection, Iteration
CST	Rights & Responsibilities		Rights & Responsibilities The Common Good Human Dignity			Participation

ST JOSEPH
Catholic Multi Academy Trust

Year 8 Computing LTP

	Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer 2
Title	Collecting and Presenting Data	Collecting and Presenting Data	Networks and the Internet	Computer Systems	Algorithms	Programming Techniques and Creating Programs
Substantive Knowledge Taught	Target Audience Collecting Data Analysing Data Presenting Data	Web Design HTML	Types of Network Wired & Wireless Networks Network Topologies	The CPU Memory Internal Storage External Storage	Decomposition Abstraction Flowcharts Pseudocode Linear Search Binary Search Bubble Sort Merge Sort	Python Data Types Operators Variables Inputs & Outputs Sequence Selection Iteration
Disciplinary Knowledge (inc. Key Concepts)	Understand how and why data is used	Principles of designs & colours that work well together	Principles of what a network is How and why networks are used	Principles of different storage types	Understanding of algorithms	Programming syntax Programming errors
Tier 3 Vocabulary	Analysis Conclusion Target Audience Manual Limitations	HTML Hyperlink Browser Formatting CSS	Network topology Wired Wireless LAN WAN	Central Internal External Storage	Decomposition Abstraction Flowcharts Pseudocode Linear Search Binary Search	Operators, Variables, Inputs and Outputs, Sequence, Selection, Iteration
CST	The Common Good Rights & Responsibilities	The Common Good Rights & Responsibilities			Participation	Participation

ST JOSEPH
Catholic Multi Academy Trust

Year 9 Computing LTP

	Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer 2
Title	Introduction to Enterprise	Programming Techniques and Creating Programs	Data Representation	Collection & Presenting Data	Networks and the Internet	Computer Systems
Substantive Knowledge Taught	Finance Marketing	Python Data Types Operators Variables Inputs & Outputs Sequence Selection Iteration	Sound Compression Logic	Databases	The Internet Network Security Threats Network Security Measures Censorship & Surveillance	Software The Operating Systems
Disciplinary Knowledge (inc. Key Concepts)	Basic business terminology Basic business finance	Programming syntax Programming errors	The purpose of logic gates How binary is used in sound and compression	How databases are used How databases can be interrogated to find information	How the internet works The difference between the world wide web and the internet	Understanding the roles of an operating system Understanding software and how used
Tier 3 Vocabulary	Finance Financial Marketing Market Research	Operators, Variables, Inputs and Outputs, Sequence, Selection, Iteration	Storage Compression Logic	Database Data	Security Threats Security Measures Censorship Surveillance	Software Hardware Operating System Defragmenting
CST	The Common Good Rights & Responsibilities Option for the poor	Participation		The Common Good	The Common Good Rights & Responsibilities Subsidiarity Human Dignity	

KS4 Overview

Computer Science

Which exam board have you chosen and why?
OCR

What is the rationale for the sequence of your KS4 curriculum?

In Year 10 students learn the content for component 1 (examination paper 1). Within year 10 there are also opportunities to be introduced to the python programming language which is the main focus on paper 2. This then moves onto a focus on component 2 (paper 2) in Year 11. To ensure that the content from paper 1 is retained, students work on a topic once every 2 weeks in Year 11.

How is your curriculum assessed? (50% EoY 10; 75% Nov Mocks; 100% March Mocks – all content should be complete by the end of SPR1 Yr 11).
End of unit assessments throughout in addition to the mock examinations mentioned above.

Component 1: Computer Systems.
80 Marks (50%), 1 Hour 30 Minutes.

> The question paper will consist of short and medium answer questions. There will also be one 8-mark extended response question. This question will enable students to demonstrate the ability to construct and develop a sustained line of reasoning.

Component 2: Computational Thinking, Algorithms and Programming.
80 Marks (50%), 1 Hour and 30 Minutes.

> Section A is worth 50 marks, and assesses students' knowledge and understanding of concepts of Computer Science. Students then apply these to problems in computational terms, where they may use an algorithmic approach.
>
> Section B is worth 30 marks, and assesses students' Practical Programming skills and their ability to design, write, test and refine programs.

The question paper will consist of short and medium answer questions.

By the end of KS4 what should all students know and be able to do? And how do you ensure this is retained in the long-term memory of students?

- Understand and apply the fundamental principles and concepts of Computer Science, including abstraction, decomposition, logic, algorithms, and data representation
- Analyse problems in computational terms through practical experience of solving such problems, including designing, writing and debugging programs
- Think creatively, innovatively, analytically, logically and critically
- Understand the components that make up digital systems, and how they communicate with one another and with other systems
- Understand the impacts of digital technology to the individual and to wider society
- Apply mathematical skills relevant to Computer Science.

ST JOSEPH
Catholic Multi Academy Trust

How does your KS4 curriculum prepare students to be successful at KS5? Enables students to look at and consider the range of courses completed at the local sixth form providers & colleges as well as at university and beyond. Links to industry and jobs mentioned throughout the course with specific links to specific topics taught.
Take one key concept or discrete piece of substantive knowledge and explain how it progresses across the two years of your curriculum. Designing, creating and refining Algorithms – Pupils will be introduced to the skill at the beginning of Year 10, it will consistently be revisited in small chunks so that students are able to continuously practice their ability to apply Computational Thinking. Following this, in Year 11 pupils will take the next step and apply their knowledge to programming to produce a variety of programs covering the programming constructs: Sequence, Selection and Iteration along with further programming techniques.
What cultural capital do students acquire from learning your curriculum? Famous computer scientists How Computer Science has helped in various areas within education, modern life (games) and war time (Bletchly Park) Ethical, legal, cultural and environmental impact of technology.

ST JOSEPH
Catholic Multi Academy Trust

Computer Science LTP

Year 10 **Computer Science**	2.4.1 Boolean Logic 1.2.3 Units 1.2.4 Data Storage (Numbers) 1.2.4 Data Storage (Characters) 1.2.4 Data Storage (Images) 1.2.4 Data Storage (Sound) 1.2.5 Compression 2.1.1 Computational Thinking 1.4.1 Threats to Computer Systems and Networks	1.1.1 Architecture of the CPU 1.1.2 CPU Performance 1.1.3 Embedded Systems 1.2.1 Primary Storage (Memory) 1.2.2 Secondary Storage 2.1.2 Designing, creating and refining algorithms	1.3.1 Networks and topologies 1.3.2 Wired and wireless networks, protocols and layers 2.1.2 Designing, creating and refining algorithms	1.4.1 Threats to computer systems and networks 1.4.2 Identifying and preventing vulnerabilities 2.1.2 Designing, creating and refining algorithms	1.5.1 Operating systems 1.5.2 Utility software 2.1.3 Searching and sorting algorithms	1.6.1 Ethical, legal, cultural and environmental impact 2.1.3 Searching and sorting algorithms
CST	The Common Good Rights & Responsibilities	The Common Good Rights & Responsibilities	The Common Good Rights & Responsibilities	The Common Good Rights & Responsibilities	The Common Good Rights & Responsibilities	The Common Good Rights & Responsibilities Human Dignity

ST JOSEPH
Catholic Multi Academy Trust

Year 11 **Computer Science**	2.5.1 Languages 2.5.2 The Integrated Development Environment (IDE) 2.2.1 Programming fundamentals 2.2.2 Data types	2.2.1 Programming fundamentals 2.2.2 Data types 2.2.3 Additional programming techniques	2.3.1 Defensive design 2.3.2 Testing	2.1.2 Designing, creating and refining algorithms 2.2.1 Programming fundamentals Revision	Revision	N/A
CST	The Common Good Rights & Responsibilities	The Common Good Rights & Responsibilities	The Common Good Rights & Responsibilities	The Common Good Rights & Responsibilities		

		Year 1			Year 2		
		Lesson 1	Lesson 2	Lesson 3	Lesson 1	Lesson 2	Lesson 3
Autumn Term							
04/09/2023	Week 1		Introduction/Admin	2.4.1 Logic Diagrams and Truth Tables	2.5.1 High and Low Level Languages	2.5.1 Translators	P1 - Component 2 Revision
11/09/2023	Week 2	2.4.1 Logic Circuits	2.4.1 Solving Problems		2.5.2 IDE's	2.2.1 Python	
18/09/2023	Week 3	2.4.1 Examination Masterclass	1.2.3 Units	1.2.4 Binary Conversion	2.2.1 Python	2.2.1 Python	P1 - Component 2 Revision
25/09/2023	Week 4	1.2.4 Binary Addition	1.2.4 Binary Shifts		2.2.1 Python	2.2.1 Python	
02/10/2023	Week 5	1.2.4 Hexadeimal	1.2.3 & 1.2.4 [N] Examination Masterclass	2.1.1 Computational Thinking	2.2.1 Python	2.2.1 Python	P1 - Component 1 Revision
09/10/2023	Week 6	1.2.4 Character Sets	1.2.4 Images		2.2.1 Python	2.2.1 Python Assessment	
16/10/2023	Week 7	1.2.4 Sound	1.2.5 Compression	2.1.2 Inputs > Processes > Outputs	Revision	Revision	P1 - Component 3 Revision
23/10/2023	Week 8	1.2.4 [C,I,S&C] Examination Masterclass	2.4.1, 1.2.3, 1.2.4 & 1.2.5 Assessment		Mock Exam	Mock Exams/Programming Challenges	
Half Term							
06/11/2023	Week 9	1.1.1 Purpose of the CPU	1.1.1 CPU Components	2.1.2 Structure Diagrams	Mock Exams/Programming Challenges	Mock Exam Review	P1 - Component 4 Revision
13/11/2023	Week 10	1.1.1 Von-Neumann Architecture	1.1.2 CPU Performance		2.2.1 Python	2.2.1 Python	
20/11/2023	Week 11	1.1.3 Embedded Systems	1.1 .1, 1.1.2 & 1.1.3 Examination Masterclass	2.1.2 Flowcharts	2.2.1 Python	2.2.1 Python	P1 - Component 5 Revision
27/11/2023	Week 12	1.1.1, 1.1.2 & 1.1.3 Assessment	1.2.1 Primary Storage		2.2.1 Python	2.2.1 Python	
04/12/2023	Week 13	1.2.2 Secondary Storage	1.2.2 Secondary Storage	2.1.2 Pseudocode	2.2.1 Python	2.2.1 Python	P1 - Component 6 Revision
11/12/2023	Week 14	1.2.1 & 1.2.2 Examination Masterclass	Revision		2.2.1 Python	2.2.1 Python Assessment	
18/12/2023	Week 15	Mock Exam	Mock Exam Review	Algorithm Practice	Python Challenges	Python Challenges	P2 - Component 4 Revision
End of Term							
Spring Term							
08/01/2024	Week 1	1.3.1 Types of Networks	1.3.1 Performance Factors and Hardware		2.3.1 Defensive Design	2.3.1 Input Validation	
15/01/2024	Week 2	1.3.1 Hardware and Client-Server/Peer to Peer	1.3.1 The Internet	Algorithm Practice	2.3.1 Maintainability	2.3.2 The Purpose of Testing	P2 - Component 1 Revision
22/01/2024	Week 3	1.3.1 Network Topologies	1.3.2 Wired and Wireless Connections		2.3.2 Test Data	2.3.2 Refining Algorithms	
29/01/2024	Week 4	1.3.2 IP and MAC Addresses	1.3.2 Protocols and Layers	Algorithm Practice	2.3.1 & 2.3.2 Examination Masterclass	Revision	P2 - Component 5 Revision
05/02/2024	Week 5	1.3.2 Encryption	1.3.1 & 1.3.2 Examination Masterclass		Mock Exam	Mock Exam	
Half Term							
19/02/2024	Week 6	1.3.1 & 1.3.2 Assessment	Review of Learning & Algorithm Practice	Algorithm Practice	Robust Programming Challenges	Robust Programming Challenges	Robust Programming Challenges
26/02/2024	Week 7	1.4.1 Forms of Attack	1.4.1 Social Engineering		Robust Programming Challenges	Robust Programming Challenges	
04/03/2024	Week 8	1.4.1 Brute Force Attacks	1.4.1 Denial of Service attacks	2.1.2 Trace Tables	Mock Exam Review	Mock Exam Review	Paper 2 Revision
11/03/2024	Week 9	1.4.1 Data Interception and Theft	1.4.1 SQL Injection		Paper 1 Revision	Paper 2 Revision	
18/03/2024	Week 10	1.4.2 Software Prevention Methods	1.4.2 Prevention Against Hacking	1.4.1 & 1.4.2 Examination Masterclass	Paper 1 Revision	Paper 2 Revision	Paper 2 Revision
25/03/2024	Week 11	1.4.1 & 1.4.2 Assessment	Review of Learning & Trace Table Practice		Paper 1 Revision	Paper 2 Revision	
End of Term							
Summer Term							
15/04/2024	Week 1	Revision	Revision	Revision	Paper 1 Revision	Paper 2 Revision	Paper 1 Revision
22/04/2024	Week 2	Revision	Revision		Paper 1 Revision	Paper 2 Revision	
29/04/2024	Week 3	Mock Exam	Mock Exams/Algorithms	2.1.3 Linear & Binary Searches	Paper 1 Revision	Paper 2 Revision	Paper 1 Revision
06/05/2024	Week 4	Mock Exams/Algorithms	Mock Exam Review		Paper 1 Revision	Paper 2 Revision	
13/05/2024	Week 5	1.5.1 Purpose of OS	1.5.1 Interfaces	2.1.3 Bubble/Merge/Insertion Sort	J277 COMPUTER SYSTEMS [PAPER 1]		
20/05/2024	Week 6	1.5.1 Memory, Multitasking & Peripherals	1.5.2 Utility Software		J277 COMPUTATIONAL THINKING, ALGORITHMS AND PROGRAMMING [PAPER 2]		
Half Term							
03/06/2024	Week 7	1.5.2 Encryption, Defrag and Compression	1.5.1 & 1.5.2 Examination Masterclass	2.1.3 Bubble/Merge/Insertion Sort			
10/06/2024	Week 8	1.5.1 & 1.5.2 Assessment					
17/06/2024	Week 9	1.6.1 Introduction & Stakeholders	1.6.1 Ethical Issues	2.1.3 Bubble/Merge/Insertion Sort			
24/06/2024	Week 10	1.6.1 Society	1.6.1 Cultural Issues			N/A	
01/07/2024	Week 11	1.6.1 Environmental Issues	1.6.1 Privacy Issues	2.1.3 Bubble/Merge/Insertion Sort			
08/07/2024	Week 12	1.6.1 Legislation	1.6.1 Software Licenses				
15/07/2024	Week 13	1.6.1 Examination Masterclass	1.6.1 Examination Masterclass	2.1.3 Examination Masterclass			
End of Term							

ST JOSEPH
Catholic Multi Academy Trust

Appendix 1: KS3 National Curriculum Review

KS3 National Curriculum (In the rows below detail the requirements of the National Curriculum – adding or subtracting rows where required).	Where is this met within your Curriculum? (Evidence the Year/Term/Title/Substantive Knowledge).
design, use and evaluate computational abstractions that model the state and behaviour of real-world problems and physical systems	Spreadsheets – Year 7 Spring 1 Databases – Year 9 Spring 1
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	Algorithms – Year 8 Summer 1
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 [for example, lists, tables or arrays]; design and develop modular programs that use procedures or functions	Programming – Year 7 Summer 2 Programming – Year 8 Autumn 2 Programming – Year 8 Summer 2 Programming – Year 9 Autumn 1 Programming – Year 9 Summer 2
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]	Data Representation – Year 7 Spring 2 Data Representation – Year 9 Spring 1
understand the hardware and software components that make up computer systems, and how they communicate with one another and with other systems	Computer Systems – Year 7 Autumn 2 Computer Systems – Year 8 Spring 1 Computer Systems – Year 8 Spring 2 Computer Systems – Year 9 Summer 1
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	Data Representation – Year 9 Spring 1

ST JOSEPH
Catholic Multi Academy Trust

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 7 – Summer 1 Year 8 – Autumn 1
create, re-use, revise and re-purpose digital artefacts for a given audience, with attention to trustworthiness, design and usability	Year 7 – Summer 1 Year 8 – Autumn 1
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 – Autumn 1 Year 8 – Autumn 2 Year 9 – Spring 2
Do you go beyond the National Curriculum? Business	Year 8 – Autumn 1 Year 9 – Autumn 2