St Paul's CATHOLIC SCHOOL

Ministerium Tuum Imple Love | Serve | Do the best that is possible

CURRICULUM OVERVIEW

Computer Science - Vision

COMPUTER SCIENCE

Vision for Curriculum

Our vision is to develop in our pupils the deep technical, problem solving, and leadership skills needed to create new computing technologies and to harness software to empower people, organisations, and society.

Intent

At St Paul's we believe Computer Science and learning talents are inseparable. The curriculum fuses Computer Science with the learning talents to develop a well-rounded pupil that is able to:

- think creatively, innovatively, analytically, logically and critically
- 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
- understand the components that make up digital systems, and how they communicate with one another and with other systems
- understand the impacts and application of digital technology to the individual and to wider society
- apply mathematical skills relevant to Computer Science.

Practical assignments set in real-life scenarios help pupils learn, build skills and develop behaviours which they will use in whatever career they choose in the future.

Implementation

We offer pupils the opportunity to study the following courses:

- Introduction to basic computer science topics for all pupils at KS3.
- GCSE Computer Science (KS4)
- A level in Computer Science (KS5)

During the teaching and learning phase pupils will experience activities such as discussions; gathering research; report writing; taking part in presentations; writing software which solves a real world problem. Through these activities pupils will develop the following skills:

- Strategic: plan; be organised; use research, reflect
- Social: collaboration; listening; working well with others; leadership
- Cognitive: imagine; ask questions; use evidence; make judgements
- Emotional: learn from mistakes; resilience; respond in depth and detail; take risks

Pupils are regularly assessed during lessons and at the end of topics to monitor progress. Pupils are also expected to monitor their own progress and set themselves personal targets to help their learning. Each pupil has a St Pauls target, which they work towards.

Impact

In addition to the realisation of the six points mentioned in the intent section above, our Computer Science curriculum provides a solid foundation for pupils who want to continue their education through applied learning and who aim to progress to further study or employment. In terms of learning talents, pupils will be able to independently: *Select and develop strategies to break down a complex problem, devise and evaluate their solutions.*

St Paul's CATHOLIC SCHOOL

Ministerium Tuum Imple Love | Serve | Do the best that is possible

CURRICULUM OVERVIEW

Computer Science - Road Map

Key Stage 3 Overview

	Year 7	Year 8	Year 9
Advent term	**Basic ICT Skills** • Introduction to computer basics • Introduction to software specific to our school: e.g class charts, OneNote. • Operating systems and file management • Word processing, spreadsheets, and presentations **Objectives:** • Develop basic computer literacy skills. • Learn to navigate operating systems and manage files effectively. • Create documents, spreadsheets, and presentations using appropriate software.	**Spreadsheets** • Introduction to spreadsheets and data organization • Formulas, functions, and charts • Using spreadsheets for data analysis **Objectives:** • Develop skills in using spreadsheets for organizing and analysing data. • Learn to create and use formulas, functions, and charts effectively.	**System Architecture** • CPU architecture (Von Neumann) • The function of the CPU (fetch-execute cycle) • Common CPU components and their functions (ALU, CU, cache) • CPU performance factors (clock speed, cache size, number of cores) **Objectives:** • Understand the architecture and function of the CPU. • Identify and describe the components of the CPU. • Explain factors affecting CPU performance.
Advent term	**Scratch** • Introduction to visual programming • Creating animations and interactive stories • Basic programming concepts (loops, conditionals, variables) • Broadcasting **Objectives:** • Understand basic programming concepts through visual programming. • Create simple animations and interactive stories using Scratch.	**Data Representation in Binary** • Understanding binary numbers and their importance in computing • Converting between binary and decimal • Representing text, images, and sound in binary **Objectives:** • Understand the basics of binary representation. • Convert between binary and decimal numbers. • Learn how data is represented in binary form.	**Memory and Storage** • The difference between RAM and ROM • The purpose and types of memory • The need for virtual memory • Common types of storage (SSD, HDD, optical, cloud) • Characteristics and suitability for different applications **Objectives:** • Differentiate between types of memory and their uses. • Understand the role of virtual memory. • Compare different storage types and their characteristics.
Lent term	**Computer Hardware & CPU** • Components of a computer system • Function and purpose of the CPU • Understanding input, output, and storage devices **Objectives:** • Identify and describe the main components of a computer. • Explain the function of the CPU and other hardware components.	**Networks** • Types of networks (LAN, WAN) • Network components (routers, switches, cables) • Understanding the internet and its structure **Objectives:** • Identify different types of networks and their components. • Understand the basic structure and function of the internet.	**Wired and Wireless Networks Network Topologies, Protocols, and Layers** • Types of networks (LAN, WAN, PAN) • The internet and its structure • Wired vs. wireless networks • Network topologies (star, mesh, bus) • Protocols (TCP/IP, HTTP, FTP) • Concept of layers (OSI model) **Objectives:** • Identify and describe different types of networks. • Explain the structure and function of the internet. • Understand various network topologies, protocols, and layers.
Lent term	**Flowol** • Introduction to flowchart programming • Controlling simulations and physical systems • Using flowcharts to solve problems **Objectives:** • Understand and create flowcharts to control simulations. • Apply flowchart programming to solve simple problems.	**Logic Gates** • Introduction to basic logic gates (AND, OR, NOT) • Creating and interpreting truth tables • Building simple logic circuits **Objectives:** • Understand the function of basic logic gates. • Use truth tables to represent logical operations. • Create simple logic circuits.	**Programming Techniques Producing Robust Programs** • Basic programming constructs (variables, data types,operators) • Control structures (loops, conditionals) • Functions and procedures • Defensive programming techniques (input validation, error handling) • Testing (syntax errors, logic errors, test plans) **Objectives:** • Understand and apply basic programming constructs. • Use control structures to develop programs. • Implement defensive programming techniques to create robust programs.
Pentecost term	**Python Turtle** • Introduction to text-based programming with Python • Drawing shapes and patterns using Turtle graphics • Basic programming concepts (loops, conditionals, functions) **Objectives:** • Learn the basics of Python programming through Turtle graphics. • Develop problem-solving skills using Python code.	**Python Programming** • Introduction to Python programming language • Variables, data types, operators, and control structures • Writing and debugging simple programs **Objectives:** • Develop programming skills using Python. • Understand and apply basic programming constructs. • Write, test, and debug Python programs.	**Programming mini project** • Application of programming concepts from last term to a case study scenario. • Case study: Subject knowledge Quiz • Learn how to document • Know the systems lifecycle: Analysis, design, implementation, testing and evaluation.
Pentecost term	**Digital Image Manipulation** • Introduction to digital images and pixels • Editing images using software (e.g., GIMP, Photoshop) • Understanding image formats and resolution **Objectives:** • Learn the basics of digital image manipulation. • Edit and create images using appropriate software tools.	**Web Development** • Basics of HTML and CSS • Creating simple web pages • Introduction to web design principles **Objectives:** • Learn the basics of web development using HTML and CSS. • Create and style simple web pages. • Understand basic web design principles.	**Artificial Intelligence** • Introduction to AI concepts • Machine learning basics • Applications of AI in various fields **Objectives:** • Understand basic concepts of artificial intelligence and machine learning. • Explore real-world applications of AI.

CATHOLIC SOCIAL TEACHING

Human Dignity · *Solidarity and Peace* · *The Common Good* · *Option for the Poor* · *Care for our Common Home* · *Rights and responibilities* · *Dignity of work*

St Paul's CATHOLIC SCHOOL

Ministerium Tuum Imple Love | Serve | Do the best that is possible

CURRICULUM OVERVIEW

Computer Science - Road Map

Key Stage 4 Overview

Year 10

Advent term

System Architecture

- CPU architecture (Von Neumann)
- The function of the CPU (fetch-execute cycle)
- Common CPU components and their functions (ALU, CU, cache)
- CPU performance factors (clock speed, cache size, number of cores)

Objectives:
- Understand the architecture and function of the CPU.
- Identify and describe the components of the CPU.
- Explain factors affecting CPU performance.

Advent term

Memory and Storage

- The difference between RAM and ROM
- The purpose and types of memory
- The need for virtual memory
- Common types of storage (SSD, HDD, optical, cloud)
- Characteristics and suitability for different applications

Objectives:
- Differentiate between types of memory and their uses.
- Understand the role of virtual memory.
- Compare different storage types and their characteristics.

Lent term

System Security and Systems Software

- Types of threats (malware, phishing, brute force attacks)
- Methods to protect systems (firewalls, anti-virus, encryption)
- Social engineering techniques
- Operating systems (functions and types)
- Utility software (defragmentation, backup, compression)
- The role of the BIOS

Objectives:
- Understand various security threats and protection methods.
- Describe the functions of operating systems and utility software.

Lent term

Algorithms

- Understanding algorithms and flowcharts
- Basic algorithms (searching and sorting: linear search, binary search, bubble sort, merge sort)
- Developing algorithms using pseudocode

Objectives:
- Explain the purpose and structure of algorithms.
- Develop and interpret basic algorithms.
- Create algorithms using pseudocode and flowcharts.

Pentecost term

Programming Techniques and Producing Robust Programs

- Basic programming constructs (variables, data types, operators)
- Control structures (loops, conditionals)
- Functions and procedures
- Defensive programming techniques (input validation, error handling)
- Testing (syntax errors, logic errors, test plans)

Objectives:
- Understand and apply basic programming constructs.
- Use control structures to develop programs.
- Implement defensive programming techniques to create robust programs.

Pentecost term

Computational Logic and Translators and Facilities of Languages

- Logic gates (AND, OR, NOT)
- Truth tables
- Boolean algebra
- Types of programming languages (high-level, low-level)
- Translators (compilers, interpreters, assemblers)
- Integrated Development Environments (IDEs)

Objectives:
- Understand and use logic gates and truth tables.
- Simplify expressions using Boolean algebra.
- Differentiate between types of programming languages and translators.

Year 11

Advent term

Data Representation

- Number systems (binary, hexadecimal, denary)
- Character encoding (ASCII, Unicode)
- Representing images and sound

Objectives:
- Convert between different number systems.
- Understand character encoding methods.
- Explain how images and sound are represented digitally.

Advent term

Review of Networks & introduction to Cyber Security

- Introduction to cyber security threats
- Methods of protection (firewalls, anti-virus, encryption)
- Understanding safe online practices

Objectives:
- Identify various cyber security threats.
- Learn methods to protect systems and data.
- Understand safe practices for online activity.

Lent term

Legal, Ethical, and Moral Issues in Computer Science

- Understanding the impact of technology on society
- Ethical considerations in technology use
- Legal issues such as copyright and data protection

Objectives:
- Discuss the societal impact of technology.
- Understand ethical and legal issues in computer science.

Lent term

Revision and Exam Preparation

- Review of all topics covered
- Past paper practice
- Exam techniques and strategies

Objectives:
- Consolidate understanding of all course topics.
- Practice and refine exam techniques.

Pentecost term

GCSES

Pentecost term

CATHOLIC SOCIAL TEACHING

Human Dignity | *Solidarity and Peace* | *The Common Good* | *Option for the Poor* | *Care for our Common Home* | *Rights and responibilities* | *Dignity of work*

St Paul's CATHOLIC SCHOOL

Ministerium Tuum Imple Love | Serve | Do the best that is possible

CURRICULUM OVERVIEW

Computer Science - Road Map

Key Stage 5 Overview

Year 12

Advent term

Components of a Computer

- Processor components
- Processor performance
- Types of processor
- Input devices
- Output devices
- Storage devices

Assessments:
- End-of-unit quiz/assessment

Computational Thinking/problem solving

- Thinking abstractly
- Thinking procedurally
- Problem recognition
- Thinking ahead
- Thinking logically & concurrently
- Problem solving

Programming Techniques
- Programming basics
- Selection
- Iteration

Assessments:
- Practical programming from booklet.

Advent term

Systems Software

- Functions of an operating system
- Types of operating system
- The nature of applications
- Programming language translators

Assessments:
- End-of-unit quiz/assessment

Programming Techniques

- Subroutines & recursion
- Use of IDE
- Use of object-oriented techniques

Programming techniques/problem solving applied to case study

Assessments:
- End-of-unit quiz/assessment
- Programming mini project

Lent term

Software Development

- Systems analysis methods
- Writing and following algorithms
- Programming paradigms
- Assembly language

Assessments:
- End-of-unit quiz/assessment

Algorithms

- Analysis and design of algorithms
- Searching algorithms
- Bubble sort & insertion sort
- Merge sort & quick sort
- Graph traversal algorithms
- Optimisation of algorithms

Assessments:
- End-of-unit quiz/assessment

Lent term

Exchanging Data

- Compression, encryption and hashing
- Database concepts
- Relational database and normalisation
- Introduction to SQL
- Defining and updating tables using SQL
- Transaction processing

Assessments:
- End-of-unit quiz/assessment

Y12 Project introduction

- Analysis
- Implementation
- Evaluation
- Design
- Testing
- Application of software methodologies/problem solving

Assessment:
Project report - analysis

Pentecost term

Networks and Web Technologies

- Structure of the internet
- Internet communication
- Network security & threats
- HTML & CSS
- Web forms & JavaScript
- Search engine indexing
- Client-server and peer-to-peer

Assessments:
- End-of-unit quiz/assessment

Y12 Project/Data structures

- Arrays, tuples, and records
- Lists and linked lists
- Hash tables
- Graphs
- Queues
- Stacks
- Trees

Assessments:
- End-of-unit quiz/assessment
- PPE
- Project report - design

Pentecost term

Data Types

- Primitive data types, binary and hexadecimal
- ASCII and Unicode
- Binary arithmetic
- Floating point arithmetic
- Bitwise manipulation and masks

Assessments:
- End-of-unit quiz/assessment

Y12 Project completion

- Completion of implementation applying data structures, algorithms, computational thinking/problem solving.

Assessment:
Project report – implementation/testing/evaluation

Year 13

Advent term

Exchanging Data

- Compression, encryption and hashing
- Database concepts
- Relational database and normalisation
- Introduction to SQL
- Defining and updating tables using SQL
- Transaction processing

Assessments:
- End-of-unit quiz/assessment

A2 Project + programming review

A2 Project + programming review
- Analysis of problem
- Implementation
- Evaluation
- OOP
- Data structures/algorithm
- Design
- Testing
- Pygame/Tkinter

Assessments:
- Analysis/design report

Advent term

Components of a Computer

- Processor components
- Processor performance
- Types of processor
- Input devices
- Output devices
- Storage devices

Assessments:
- End-of-unit quiz/assessment

A2 Project + programming review + PPE Prep

- Implementation
- Evaluation
- OOP
- Data structures/algorithm
- Testing
- Pygame/Tkinter

Assessments:
- A2 project report
- Y13 PPE

Lent term

Systems Software

- Functions of an operating system
- Types of operating system
- The nature of applications
- Programming language translators

Assessments:
- End-of-unit quiz/assessment

Legal, Moral, Ethical and Cultural Issues

- Computing related legislations
- Ethical, moral, and cultural issues
- Privacy and censorship

A2 Project – improvements

Lent term

Networks and Web Technologies

- Structure of the internet
- Internet communication
- Network security & threats
- HTML & CSS
- Web forms & JavaScript
- Search engine indexing
- Client-server and peer-to-peer

Assessments:
- End-of-unit quiz/assessment

Exam prep – weak topics based on PPE and classroom assessment

- Data structures
- Algorithms
- Data Types
- Components of a processor

Pentecost term

Exam prep – weak topics

- Student request + identified through assessments

Exam prep – weak topics

Student request + identified through assessments

Pentecost term

EXAMS

CATHOLIC SOCIAL TEACHING

- Human Dignity
- Solidarity and Peace
- The Common Good
- Option for the Poor
- Care for our Common Home
- Rights and responibilities
- Dignity of work