

# National Curriculum Objectives

## Key Stage One:

Understand what algorithms are; how they are implemented as programs on digital devices; and that programs execute by following precise and unambiguous instructions.

Create and debug simple programs.

Use logical reasoning to predict the behaviour of simple programs.

Use technology purposefully to create, organise, store, manipulate and retrieve digital content.

Recognise common uses of information technology beyond school.

Use technology safely and respectfully, keeping personal information private; identify where to go for help and support when they have concerns about content or contact on the internet or other online technologies.

## Key Stage Two:

Design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts.

Use sequence, selection, and repetition in programs; work with variables and various forms of input and output.

Use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs.

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.

Use search technologies effectively, appreciate how results are selected and ranked, and be discerning in evaluating digital content.

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.

Use technology safely, respectfully and responsibly; recognise acceptable and unacceptable behaviour; identify a range of ways to report concerns about content and contact.

## Key Stage Three:

Design, use and evaluate computational abstractions that model the state and behaviour of real-world problems and physical systems.

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.

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.

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].

Understand the hardware and software components that make up computer systems, and how they communicate with one another and with other systems.

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.

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.

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.

# Computer Science

# St. Joseph's Catholic Academy Computing Curriculum Progression of Skills

<p><b>Key Stage One</b></p> <ul style="list-style-type: none"> <li>Understand what algorithms are; how they are implemented as programs on digital devices; and that programs execute by following precise and unambiguous instructions.</li> <li>Create and debug simple programs.</li> <li>Use logical reasoning to predict the behaviour of simple programs.</li> </ul> <p><b>Key Stage Two</b></p> <ul style="list-style-type: none"> <li>Design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts.</li> <li>Use sequence, selection and repetition in programs; work with variables and various forms of input and output.</li> <li>Use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs.</li> <li>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.</li> </ul>				<p><b>Key Stage Three</b></p> <ul style="list-style-type: none"> <li>Design, use and evaluate computational abstractions that model the state and behaviour of real-world problems and physical systems.</li> <li>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.</li> <li>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.</li> <li>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].</li> <li>Understand the hardware and software components that make up computer systems, and how they communicate with one another and with other systems.</li> <li>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.</li> </ul>		
EYFS	Year 1	Year 2	Year 3	Year 4	Year 5	Year 6
<p>Be confident to try new activities and show independence, resilience and perseverance in the face of challenge (ELG: Personal, Social and Emotional Development – Managing Self)</p> <p>Explore how things work (Development Matters: Understanding the World – Three and Four-Year-Olds).</p> <p>Show resilience and perseverance in the face of a challenge (Development Matters: Personal, Social and Emotional Development - Reception).</p> <p>Explore, use and refine a variety of artistic effects to express their ideas and feelings (Development Matters: Expressive Arts and Design - Reception).</p> <p>Safely use and explore a variety of materials, tools and techniques, experimenting with colour, design, texture, form and function (ELG: Expressive Arts &amp; Design – Creating with Materials).</p>	<p>Understand that an algorithm is a set of instructions used to solve a problem or achieve an objective.</p> <p>Know that a computer program turns an algorithm into code that the computer can understand.</p> <p>Work out what is wrong with a simple algorithm when the steps are out of order.</p> <p>Know that an unexpected outcome is due to the code they have created and can make logical attempts to fix the code.</p> <p>When looking at a program, read code one line at a time and make good attempts to envision the bigger picture of the overall effect of the program.</p>	<p>Explain that an algorithm is a set of instructions to complete a task.</p> <p>When designing simple programs, show an awareness of the need to be precise with algorithms so that they can be successfully converted into code.</p> <p>Create a simple program that achieves a specific purpose.</p> <p>Identify and correct some errors.</p> <p>Display a growing awareness of the need for logical, programmable steps through program design.</p> <p>Identify the parts of a program that respond to specific events and initiate specific actions. For example, write a cause and effect sentence of what will happen in a program.</p>	<p>Turn a simple real-life situation into an algorithm for a program by deconstructing it into manageable parts.</p> <p>Through program design, show thought of the desired task and how this translates into code.</p> <p>Identify an error within a program that prevents it following the desired algorithm and then fix it.</p> <p>Demonstrate the ability to design and code a program that follows a simple sequence.</p> <p>Experiment with timers to achieve repetition effects in programs.</p> <p>Begin to understand the difference in the effect of using a timer command rather than a repeat command when creating repetition effects.</p> <p>Through program design, show consideration of the structure of a program in logical, achievable steps and absorb some new knowledge of coding structures. For example, repetition and use of timers.</p> <p>Make good attempts to 'step through' more complex code in order to identify errors in algorithms and correct this - e.g. in programs such as Logo, 'read' programs with</p>	<p>When turning a real-life situation into an algorithm, through design, show consideration of the required task and how to accomplish this in code using coding structures for selection and repetition.</p> <p>Make more intuitive attempts to debug own programs.</p> <p>Show use of timers to achieve repetition effects are becoming more logical and are integrated into program designs.</p> <p>Understand 'IF statements' for selection and attempt to combine these with other coding structures including variables to achieve the effects that they design in their programs.</p> <p>As well as understanding how variables can be used to store information while a program is executing, use and manipulate the value of variables.</p> <p>Make use of user inputs and outputs such as 'print to screen'. e.g. 2Code.</p> <p>Through program design, show consideration of the structure of a program in logical, achievable steps and absorb some new knowledge of coding structures. For example, 'IF' statements, repetition and variables.</p>	<p>Attempt to turn more complex real-life situations into algorithms for a program by deconstructing it into manageable parts.</p> <p>Test and debug their programs regularly and use logical methods to identify the approximate cause of any bug (but may need some support identifying the specific line of code).</p> <p>Translate algorithms that include sequence, selection and repetition into code with increasing ease and designs show consideration of how to accomplish the set task in code utilising such structures.</p> <p>Combine sequence, selection and repetition with other coding structures to achieve algorithm design.</p> <p>During coding, begin to think about code structure in terms of the ability to debug and interpret the code later, e.g. the use of tabs to organise code and the naming of variables.</p> <p>Understand the value of computer networks but also aware of the main dangers.</p> <p>Recognise what personal information is and explain how this can be kept safe.</p>	<p>Turn a more complex programming task into an algorithm by identifying the important aspects of the task (abstraction) and then decomposing them in a logical way using knowledge of possible coding structures and applying skills from previous programs.</p> <p>Test and debug programs regularly and use logical methods to identify the cause of bugs, demonstrating a systematic approach to try to identify a particular line of code causing a problem.</p> <p>Translate algorithms that include sequence, selection and repetition into code and designs show consideration of how to accomplish the set task in code utilising such structures, including nesting structures within each other.</p> <p>Display an improving understanding of variables in coding, outputs such as sound and movement, inputs from the user of the program such as button clicks and the value of functions.</p> <p>Interpret a program in parts and make logical attempts to put the separate parts of a complex</p>

## St. Joseph's Catholic Academy Computing Curriculum Progression of Skills

			<p>several steps and predict the outcome accurately.</p> <p>List a range of ways that the Internet can be used to provide different methods of communication.</p> <p>Use some of these methods of communication, e.g. being able to open, respond to and attach files to emails using 2Email.</p> <p>Describe appropriate email conventions when communicating in this way.</p>	<p>Trace code and use step-through methods to identify errors in code and make logical attempts to correct this.</p> <p>In programs such as Logo, 'read' programs with several steps and predict the outcome accurately</p> <p>Recognise the main component parts of hardware which allow computers to join and form a network.</p> <p>Improve understanding of the online safety implications associated with the ways the internet can be used to provide different methods of communication.</p>	<p>Select the most appropriate form of online communications contingent on audience and digital content, e.g. 2Blog, 2Email, Display Boards.</p>	<p>algorithm together to explain the program as a whole.</p> <p>Understand and explain in some depth the difference between the internet and the World Wide Web.</p> <p>Know what a WAN and LAN are and describe how they access the Internet in school.</p>
--	--	--	---	---	--	--

## St. Joseph's Catholic Academy Computing Curriculum Progression of Skills

# Information Technology

### Key Stage One

- Use technology purposefully to create, organise, store, manipulate and retrieve digital content.

### Key Stage Two

- Use search technologies effectively, appreciate how results are selected and ranked, and be discerning in evaluating digital content.
- 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.

### Key Stage Three

- 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

EYFS	Year 1	Year 2	Year 3	Year 4	Year 5	Year 6
<p>Explore how things work (Development Matters: Understanding the World – Three and Four-Year-Olds).</p> <p>Safely use and explore a variety of materials, tools and techniques, experimenting with colour, design, texture, form and function (ELG: Expressive Arts &amp; Design – Creating with Materials).</p>	<p>Sort, collate, edit and store simple digital content e.g. name, save and retrieve work and follow simple instructions to access online resources, use Purple Mash 2Quiz example (sorting shapes), 2Code design mode (manipulating backgrounds) or use pictogram software such as 2Count.</p>	<p>Demonstrate an ability to organise data using, for example, a database such as 2Investigate and retrieve specific data for conducting simple searches.</p> <p>Edit more complex digital data such as music compositions within 2Sequence.</p> <p>Confidently create, name, save and retrieve content.</p> <p>Use a range of media in digital content including photos, text and sound.</p>	<p>Carry out simple searches to retrieve digital content.</p> <p>Understand that to do this, one must connect to the internet and use a search engine such as Purple Mash search or internet-wide search engines.</p> <p>Collect, analyse, evaluate and present data and information using a selection of software, e.g. using a branching database (2Question), using software such as 2Graph.</p> <p>Consider what software is most appropriate for a given task.</p> <p>Create purposeful content to attach to emails, e.g. 2Respond.</p>	<p>Understand the function, features and layout of a search engine.</p> <p>Appraise selected webpages for credibility and information at a basic level.</p> <p>Make improvements to digital solutions based on feedback.</p> <p>Make informed software choices when presenting information and data.</p> <p>Create linked content using a range of software such as 2Connect and 2Publish+.</p> <p>Share digital content within their community, i.e. using Virtual Display Boards.</p>	<p>Search with greater complexity for digital content when using a search engine.</p> <p>Explain in some detail how credible a webpage is and the information it contains.</p> <p>Make appropriate improvements to digital solutions based on feedback received and confidently comment on the success of the solution - e.g. creating a program to meet a design brief using 2Code.</p> <p>Objectively review solutions from others.</p> <p>Collaboratively create content and solutions using digital features within software such as collaborative mode.</p> <p>Use several ways of sharing digital content, i.e. 2Blog, Display Boards and 2Email.</p>	<p>Apply filters when searching for digital content.</p> <p>Explain in detail how credible a webpage is and the information it contains.</p> <p>Compare a range of digital content sources and rate them in terms of content quality and accuracy.</p> <p>Critical thinking skills in everyday use of online communication.</p> <p>Make clear connections to the audience when designing and creating digital content.</p> <p>Design and create blogs to become a content creator on the Internet, e.g. 2Blog.</p> <p>Use criteria to evaluate the quality of digital solutions and are able to identify improvements, making some refinements.</p>

## St. Joseph's Catholic Academy Computing Curriculum Progression of Skills

# Digital Literacy

### Key Stage One

- Recognise common uses of information technology beyond school.
- Use technology safely and respectfully, keeping personal information private; identify where to go for help and support when they have concerns about content or contact on the internet or other online technologies.

### Key Stage Two

- Use technology safely, respectfully and responsibly; recognise acceptable/ unacceptable behaviour; identify a range of ways to report concern about content and contact.

### Key Stage Three

- 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

EYFS	Year 1	Year 2	Year 3	Year 4	Year 5	Year 6
<p>Increasingly follow rules, understanding why they are important (Development Matters: Personal, Social and Emotional Development – Three and Four-Year-Olds).</p> <p>Explain the reasons for rules, know right from wrong and try to behave accordingly (ELG: Personal, Social and Emotional Development – Managing Self).</p>	<p>Understand what is meant by technology and identify a variety of examples both in and out of school.</p> <p>Make a distinction between objects that use modern technology and those that do not e.g. a microwave vs. a chair.</p> <p>Understand the importance of keeping information, such as usernames and passwords, private and actively demonstrate this in lessons.</p> <p>Take ownership of work and save this in own private space such as My Work folder on Purple Mash.</p>	<p>Effectively retrieve relevant, purposeful digital content using a search engine.</p> <p>Apply learning of effective searching beyond the classroom.</p> <p>Share this knowledge, e.g. 2Publish example template.</p> <p>Make links between technology seen all around, coding and multimedia work completed in school e.g. animations, interactive code and programs.</p> <p>Know the implications of inappropriate online searches.</p> <p>Begin to understand how things are shared electronically such as posting work to the Purple Mash display board.</p> <p>Develop an understanding of using email safely by using 2Respond activities on Purple Mash and know ways of reporting inappropriate behaviours and content to a trusted adult.</p>	<p>Demonstrate the importance of having a secure password and not sharing this with anyone else.</p> <p>Explain the negative implications of failure to keep passwords safe and secure.</p> <p>Understand the importance of staying safe and the importance of own conduct when using familiar communication tools such as 2Email in Purple Mash.</p> <p>Know more than one way to report unacceptable content and contact.</p>	<p>Explore key concepts relating to online safety using concept mapping such as 2Connect.</p> <p>Help others to understand the importance of online safety.</p> <p>Know a range of ways of reporting inappropriate content and contact.</p>	<p>Have a secure knowledge of common online safety rules and can apply this by demonstrating the safe and respectful use of a few different technologies and online services.</p> <p>Implicitly relate appropriate online behaviour to their right to personal privacy and mental wellbeing of all.</p>	<p>Demonstrate the safe and respectful use of a range of different technologies and online services.</p> <p>Identify more discreet inappropriate behaviours through developing critical thinking, e.g. 2Respond activities.</p> <p>Recognise the value in preserving their privacy when online for the safety of all.</p>