Progression Overview

	EYFS	Year 1	Year 2	Year 3	Year 4	Year 5	Year 6
Digital Literacy 	-Understand basic online safety, such as not sharing personal information. -Learn about who to talk to if they feel unsafe online. -Begin to explore safe online activities using child-friendly websites.	-Understand what is meant by technology and can identify a variety of examples both in and out of school. -Make a distinction between objects that use modern technology and those that do not e.g. a microwave vs. a chair. -Understand the importance of keeping information, such as their usernames and passwords, private and actively demonstrate this in lessons. -Take ownership of their work and save this in their own private space such as their My Work folder on Purple Mash.	-Effectively retrieve relevant, purposeful digital content using a search engine. -Apply their learning of effective searching beyond the classroom. -Share this knowledge, e.g. 2Publish example template. Children make links between technology they see around them, coding and multimedia work they do in school e.g. animations, interactive code and programs. -Know the implications of inappropriate online searches. -Begin to understand how things are shared electronically such as posting work to the Purple Mash display board. -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.	-Demonstrate the importance of having a secure password and not sharing this with anyone else. -Explain the negative implications of failure to keep passwords safe and secure. -Understand the importance of staying safe and the importance of their conduct when using familiar communication tools such as 2Email in Purple Mash. -Know more than one way to report unacceptable content and contact.	-Explore key concepts relating to online safety using concept mapping such as 2Connect. -Help others to understand the importance of online safety. -Know a range of ways of reporting inappropriate content and contact.	-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. -Implicitly relate appropriate online behaviour to their right to personal privacy and mental wellbeing of themselves and others.	-Demonstrate the safe and respectful use of a range of different technologies and online services. They identify more discreet inappropriate behaviours through developing critical thinking, e.g. 2Respond activities. -Recognise the value in preserving their privacy when online for their own and other people's safety.
	EYFS	Year 1	Year 2	Year 3	Year 4	Year 5	Year 6

Information Technology 	-Recognise that technology is used in everyday life (e.g. tablets, phones, smart speakers, ATMs) -Explore different types of technology (e.g. keyboards, touchscreens, programmable toys). -Explore digital drawing and simple animations.	-Sort, collate, edit and store simple digital content e.g. children can name, save and retrieve their work and follow simple instructions to access online resources, use Purple Mash 2Quiz example (sorting shapes), 2Code design mode (manipulating backgrounds) or using pictogram software such as 2Count.	-Demonstrate an ability to organise data using, for example, a database such as 2Investigate and can retrieve specific data for conducting simple searches. - Edit more complex digital data such as music compositions within 2Sequence. -Show confidence when creating, naming, saving and retrieving content. -Use a range of media in their digital content including photos, text and sound.	-Carry out simple searches to retrieve digital content. -Understand that to do this, they are connecting to the internet and using a search engine such as Purple Mash search or internet-wide search engines. -Collect, analyse, evaluate and present data and information using a selection of software, e.g. using a branching database (2 Question), using software such as 2 Graph. -Consider what software is most appropriate for a given task. -Create purposeful content to attach to emails, e.g. 2Respond.	-Understand the function, features and layout of a search engine. -Appraise selected webpages for credibility and information at a basic level. . -Make improvements to digital solutions based on feedback. -Make informed software choices when presenting information and data. -Create linked content using a range of software such as 2Connect and 2Publish+. -Share digital content within their community, i.e. using Virtual Display Boards.	-Search with greater complexity for digital content when using a search engine. -Explain in some detail how credible a webpage is and the information it contains. -Make appropriate improvements to digital solutions based on feedback received and can confidently comment on the success of the solution. e.g. creating their own program to meet a design brief using 2Code. -Objectively review solutions from others. -Collaboratively create content and solutions using digital features within software such as collaborative mode. -Use several ways of sharing digital content, i.e. 2Blog, Display Boards and 2Email.	-Readily apply filters when searching for digital content. -Explain in detail how credible a webpage is and the information it contains. -Compare a range of digital content sources and rate them in terms of content quality and accuracy. -Use critical thinking skills in everyday use of online communication. -Make clear connections to the audience when designing and creating digital content. -Design and create their own blogs to become a content creator on the internet, e.g. 2Blog. -Use criteria to evaluate the quality of digital solutions and identify improvements, making some refinements.
	EYFS	**Year 1**	**Year 2**	**Year 3**	**Year 4**	**Year 5**	**Year 6**
Computer Science 	-Develop problem-solving skills through sequencing, patterns and logical thinking. -Use unplugged activities and Beebots to learn about algorithms	-Understand that an algorithm is a set of instructions used to solve a problem or achieve an objective. -Know that a computer program turns an algorithm	Explain that an algorithm is a set of instructions to complete a task. -When designing simple programs, show an awareness of the need to be precise with their	-Turn a simple real-life situation into an algorithm for a program by deconstructing it into manageable parts. Their design	-When turning a real-life situation into an algorithm, show that they are thinking of the required task and how to accomplish this in code using coding structures for selection and repetition.	Attempt to turn more complex real-life situations into algorithms for a program by deconstructing it into manageable parts. Test and debug their programs as they go	-Turn a more complex programming task into an algorithm by identifying the important aspects of the task

	(e.g. Robot instructions game) -Learn to give and follow simple instructions (precursors to coding)	into code that the computer can understand. -Work out what is wrong with a simple algorithm when the steps are out of order, e.g. The Wrong Sandwich in Purple Mash and can write their own simple algorithm, e.g. Colouring in a Bird activity. -Know that an unexpected outcome is due to the code they have created and can make logical attempts to fix the code, e.g. Bubbles activity in 2Code. -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. For example, interpret where the turtle in 2Go challenges will end up at the end of the program.	algorithms so that they can be successfully converted into code. -Create a simple program that achieves a specific purpose. -Identify and correct some errors, e.g. Debug Challenges: Chimp. - Display a growing awareness of the need for logical, programmable steps when designing programs. -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.	shows that they are thinking of the desired task and how this translates into code. Identify an error within their program that prevents it following the desired algorithm and then fix it. -Demonstrate the ability to design and code a program that follows a simple sequence. -Experiment with timers to achieve repetition effects in their programs. -Begin to understand the difference in the effect of using a timer command rather than a repeat command when creating repetition effects. -Design programs to show that they are thinking of the structure of a program in logical, achievable steps and absorbing some new knowledge of coding structures. For example, repetition and use of timers.	-Make more intuitive attempts to debug their own programs. -Use timers to achieve repetition effects are becoming more logical and are integrated into their program designs. -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. -As well as understanding how variables can be used to store information while a program is executing, use and manipulate the value of variables. -Make use of user inputs and outputs such as 'print to screen'. e.g. 2Code. -Show in their designs that they are thinking of the structure of a program in logical, achievable steps and absorbing some new knowledge of coding structures. For example, 'IF' statements, repetition and variables. -Trace code and use step-through methods to identify errors in code and make logical attempts to correct this. -In programs such as Logo, 'read' programs with several steps and predict the outcome accurately. -Recognise the main component parts of hardware which allow	and can use logical methods to identify the approximate cause of any bug but may need some support identifying the specific line of code. -Translate algorithms that include sequence, selection and repetition into code with increasing ease and own designs show that they are thinking of how to accomplish the set task in code utilising such structures. -Combine, sequence, selection and repetition with other coding structures to achieve their algorithm design. -Begin to think about their 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. -Understand the value of computer networks but are also aware of the main dangers. -Recognise what personal information is and explain how this can be kept safe. -Select the most appropriate form of online communications contingent on audience and digital content, e.g. 2Blog, 2Email, Display Boards.	(abstraction) and then decomposing them in a logical way using knowledge of possible coding structures and applying skills from previous programs. -Test and debug their program as they go 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. -Translate algorithms that include sequence, selection and repetition into code and their own designs show thinking of how to accomplish the set task in code utilising such structures, including nesting structures within each other. -Coding displays an improving understanding of variables in

				-Make good attempts to 'step through' more complex code in order to identify errors in algorithms and can correct this. e.g. In programs such as Logo, -'Read' programs with several steps and predict the outcome accurately. -List a range of ways that the Internet can be used to provide different methods of communication. -Use some methods of communication, e.g. being able to open, respond to and attach files to emails using 2Email. -Describe appropriate email conventions when communicating in this way.	computers to join and form a network. -Understand the online safety implications associated with the ways the Internet can be used to provide different methods of communication.		coding, outputs such as sound and movement, inputs from the user of the program such as button clicks and the value of functions. Interpret a program in parts and make logical attempts to put the separate parts of a complex algorithm together to explain the program as a whole.

Examples of Progression in Computing at Our Lady of the Angels

Coding
In the Early Years, children begin their progression in coding by learning to programme a floor robot, such as a bee-bot, to follow a simple instruction that allows it to follow a programmed route. Children also begin to learn how to interpret a simple set of instructions, allowing them to predict an outcome. Throughout years 1 and 2, children begin to learn more complex vocabulary such as algorithm and debug. They use this vocabulary to support them to create their own computer programmes using 2Code, and then to identify problems with a code and debug the programme to make it work. Within years 3 and 4, children are introduced to more complex coding skills such as designing and creating interactive scenes and then their own games. In order to do this, children in year 3 use different types of timers to create different effects and children in year 4 learn that if/else statements can test the conditions of a computer programme and work accordingly. Throughout years 5 and 6, children are introduced to simulation and how they can link computer programmes to real life skills. This is then translated back into 2Code, where children design their own playable computer games using simulation. Children in year 6 also learn to include text within their game to create a text-adventure game.
Online Safety
In the Early Years, children begin their learning about online safety by learning about the importance of not sharing personal information, recognizing safe online environments, and identifying trusted adults to turn to when feeling unsafe. In Year 1 their exploration expands into identifying and distinguishing technology in various forms, fostering awareness of the role it plays in their lives. In Year 2 learners build essential skills like maintaining privacy through secure passwords, practicing responsible online conduct, and safely using communication tools like email. Year 3 builds on this by teaching pupils to understand the implications of inappropriate behavior or unsafe practices online and this further develops in Year 4 where pupils are taught to critically evaluate digital content and apply effective search techniques. In Year 5 whilst carrying out tasks such as coding, animations, and multimedia projects, learners make connections between their activities and broader digital concepts. As their understanding deepens in Year 6, children begin to advocate for online safety and respectful use of digital resources, recognizing the impact of their behavior on their privacy and emotional well-being, as well as that of others. This progression develops not just technical skills but also responsibility, resilience, and a sense of stewardship in their digital interactions.