



## St. Stephen's Junior School

### Curriculum Overview: Computing

#### Curriculum Intent:

At St. Stephen's Junior, we believe strongly in the importance of equipping our children for the world around them.

To ensure that pupils develop a secure knowledge that they can build on, our Computing curriculum is organised into a progression model that outlines the skills, knowledge and vocabulary to be taught in a sequentially coherent way. The key strands are: Computer Science, Digital Literacy and Information Technology. These are all sequenced to ensure that pupils build on secure prior knowledge, taught fortnightly through a range of units involving applications, the internet and specific programmes linked to the teaching.

With the aim of preparing our pupils to succeed in a future world that is becoming more reliant on technology, we strive to provide the children with a diverse Computing curriculum that can be adapted in line with technological society changes.

We encourage and support our pupils to become sensible users of technology and conscientious members of the global online community, we educate them on how to be responsible and safe when accessing the internet.

Through our Computing curriculum we aim for our children leave St Stephen's as confident and adaptable learners who are computer literate.



## Progression in learning:

To ensure progression in development of knowledge and the important skills associated with Computing, each child will spend time developing their understanding of Information Technology, Computer Science and Digital Literacy. Children will be taught Computing weekly and will have the use of iPads and Laptops to complete these lessons. Children will experience programming, processing, sorting data, creating content, animation and internet safety. Each of these units will be sorted into achievable weekly objectives and revisited regularly for the children to build upon and develop their knowledge and understanding.

An example of this progression is shown below:

Year 3	Year 4	Year 5	Year 6
<b>iProgram</b> <ul style="list-style-type: none"> <li>☞ To program an animation that executes a sequence of statements</li> <li>☞ To understand that computer programs containing graphics use x y coordinates and turns are measured in degrees</li> <li>☞ To program a sequence of instructions that create visual effects</li> <li>☞ To import, create and record sounds</li> <li>☞ To understand that algorithms and programs can involve repetition</li> <li>☞ To predict the outcome of a simple algorithm</li> <li>☞ To combine images, sounds and movement to create a personal animation</li> </ul>	<b>iProgram</b> <ul style="list-style-type: none"> <li>☞ To understand that a program is a sequence of statements written in a programming language</li> <li>☞ To program a sequence of statements</li> <li>☞ To program an object to move and draw</li> <li>☞ To understand that commands and actions can be programmed to be executed depending upon whether a condition is true or not</li> <li>☞ To combine repetition and conditional statements in a program</li> </ul>	<b>iProgram</b> <ul style="list-style-type: none"> <li>☞ To understand that computer programs containing graphics use x y coordinates and turns are measured in degrees</li> <li>☞ To use conditional (if) statements</li> <li>☞ To understand that some variables can only be true or false (boolean)</li> <li>☞ To understand that programs can do different things if the value of a boolean variable is true or false (conditional statements)</li> <li>☞ To use variables in programs</li> </ul>	<b>iProgram</b> <ul style="list-style-type: none"> <li>☞ To program a computer game by sequencing conditional statements</li> <li>☞ To use variables in programs</li> <li>☞ To use procedures in programs</li> <li>☞ To understand that the behaviour of a computer program should be planned</li> <li>☞ To understand that programs are developed according to a plan</li> <li>☞ To develop strategies for testing and debugging computer programs</li> </ul>