# Medium Term Plan - Computing

| Term / Class | EYFS | Year 1 | Year 2 | Year 3 | Year 4 | Year 5 | Year 6 |
|---|---|---|---|---|---|---|---|
| **Pentecost 2** | **Water: Seasides and Holidays**<br><br>**eSafety objective**<br>Know that they should ask an adult before clicking unfamiliar buttons, videos or games online.<br>**Computing objectives**<br>1. Use simple drawing or painting apps to create seaside pictures.<br>2. Explore digital stories or videos about holidays and the seaside.<br>3. Use touch controls confidently on a device.<br>4. Follow simple instructions to complete a digital activity.<br>5. Talk about favourite technology they use at home or school.<br>**Assessment**<br>• **Observation of device control**<br>• **Can children follow instructions independently?**<br>• **Can children talk about safe technology use?** | **Robots**<br>**Programming**<br>**eSafety objective**<br>Know to tell an adult if a program or game shows something unexpected or upsetting.<br>**Computing objectives**<br>1. Give simple instructions to a programmable toy or robot.<br>2. Create a sequence of commands to complete a task.<br>3. Predict what a robot will do from a set of instructions.<br>4. Debug simple errors in a program.<br>5. Explain how instructions must be clear and in the correct order.<br>**Assessment**<br>• **Can children program a simple route?**<br>• **Can they spot and fix mistakes?**<br>• **Can they explain what their algorithm does?** | **Buckets and Spades**<br>**Topic: Programming**<br>• **eSafety objective**<br>• Understand that only trusted adults should help them download or use new apps and games.<br>**Computing objectives**<br>• Create simple programs using sequences of instructions.<br>• Use logical reasoning to predict outcomes of programs.<br>• Debug programs when they do not work correctly.<br>• Use repetition in simple programs.<br>• Explain how programs follow precise instructions.<br>• **Assessment**<br>• Can children predict outcomes correctly?<br>• Can they debug independently?<br>• Can they create a simple working program? | **How does your garden grow?**<br>**Topic: Presenting Information**<br>**eSafety objective**<br>Know that information shared online should be respectful and suitable for the audience.<br>**Computing objectives**<br>1. Create a simple outline of a digital presentation about plants or gardens.<br>2. Combine text and images effectively.<br>3. Use slide layouts and formatting tools clearly.<br>4. Organise information for a chosen audience.<br>5. Present information confidently using digital tools.<br>**Assessment**<br>• **Quality of presentation layout**<br>• **Appropriate use of text/images**<br>• **Ability to explain information clearly** | **Hunted**<br>**Topic: Computational Thinking**<br>**eSafety objective**<br>Recognise that not all online games or challenges are safe and know how to report concerns.<br>**Computing objectives**<br>1. Break problems into smaller steps (decomposition).<br>2. Create algorithms to solve problems.<br>3. Use logical reasoning to predict outcomes.<br>4. Identify patterns and repetition in tasks.<br>5. Debug algorithms and improve solutions.<br>**Assessment**<br>• **Can children decompose a task?**<br>• **Can they explain their algorithm?**<br>• **Can they debug effectively?** | **Food, Glorious Food!**<br>**Topic: Programming**<br>**eSafety objective**<br>Understand how to communicate respectfully when sharing projects or commenting online.<br>**Computing objectives**<br>1. Design and create programs using sequence, selection and repetition.<br>2. Use variables in programs.<br>3. Debug programs systematically.<br>4. Evaluate and improve programs for a purpose.<br>5. Explain how their program works.<br>**Assessment**<br>• **Working use of variables and selection**<br>• **Ability to debug independently**<br>• **Quality of final program** | **Oh I do like to be beside the seaside (cont.)**<br><br>**Computing objectives**<br>6. Use inputs and outputs effectively within their program (e.g. keyboard controls, score counters, sound effects or animations).<br>7. Design programs with a clear audience and purpose in mind, making choices about layout, instructions and usability.<br>8. Use decomposition to break a larger programming task into smaller manageable parts before coding.<br>9. Compare different coding solutions and explain which is most effective and why.<br>10. Present their finished project to others, explaining the coding techniques, challenges and improvements made during development.<br><br>**Assessment - Functionality of program**<br>**Debugging ability** |