# Second Term Programming Scheme

## Grade 1

|  |  |  |  |
| --- | --- | --- | --- |
| Week | Topic | Activities | Objectives |
| 1 | Review of First-Term Topics | Discussion and practice of key concepts like mouse and keyboard use. | Reinforce foundational computer skills. |
| 2 | Understanding Sequences (Continuation) | More hands-on exercises with sequences in ScratchJr. | Strengthen understanding of sequences. |
| 3 | Conditional Statements in ScratchJr | Introduce basic conditions (if-then statements). | Develop logical thinking skills. |
| 4 | Introduction to Events | Explain events in programming. | Understand how actions trigger responses in programs. |
| 5 | Adding Sounds and Music (Review) | More complex sound additions in ScratchJr. | Enhance creativity in projects. |
| 6 | Advanced Loops | Explore nested loops with examples. | Improve problem-solving with loops. |
| 7 | Debugging Basics (Review and Expand) | More debugging challenges. | Identify and fix errors systematically. |
| 8 | Final Project Preparation | Plan a new project incorporating learned concepts. | Apply all learned skills. |
| 9 | Final Project Completion | Complete and refine projects. | Strengthen project development skills. |
| 10 | Presentation and Feedback | Showcase projects and give peer reviews. | Develop communication and collaboration. |

## Grade 2

|  |  |  |  |
| --- | --- | --- | --- |
| Week | Topic | Activities | Objectives |
| 1 | Review of First-Term Topics | Recap computer basics, mouse, and keyboard skills. | Solidify prior learning. |
| 2 | Advanced Commands in ScratchJr | More practice with complex commands. | Improve programming fluency. |
| 3 | Introduction to Loops | Teach loops using ScratchJr activities. | Build logical thinking with repetition. |
| 4 | Advanced Events | Create event-driven programs. | Understand how user actions affect programs. |
| 5 | Conditional Logic (Review and Expand) | Work on multi-condition programs. | Improve decision-making in programming. |
| 6 | Debugging Techniques (More Complex) | Find and fix complex errors. | Develop problem-solving skills. |
| 7 | Adding Backgrounds and Effects | Introduce more customization options. | Enhance creativity in projects. |
| 8 | Final Project Planning | Brainstorm and start project work. | Encourage independent project thinking. |
| 9 | Final Project Development | Work on individual projects. | Apply all programming skills. |
| 10 | Project Presentations | Showcase projects to class. | Gain confidence in coding and presenting. |

## Grade 3

|  |  |  |  |
| --- | --- | --- | --- |
| Week | Topic | Activities | Objectives |
| 1 | Review of First-Term Topics | Recap programming fundamentals. | Reinforce knowledge. |
| 2 | Scratch Basics and Animations | Create animations using Scratch. | Improve storytelling and creativity. |
| 3 | Interactive Scratch Projects | Develop interactive stories or quizzes. | Strengthen logical thinking. |
| 4 | Game Development with Scratch | Design simple games. | Enhance problem-solving and creativity. |
| 5 | Advanced Scratch Features | Work with clones, messages, and variables. | Develop advanced Scratch skills. |
| 6 | Debugging Techniques | Identify and fix common errors. | Strengthen problem-solving abilities. |
| 7 | Final Project Planning | Plan a Scratch-based project. | Apply all learned concepts. |
| 8 | Project Development | Work on individual or group projects. | Enhance coding independence. |
| 9 | Final Project Refinement | Debug and refine projects. | Improve project quality. |
| 10 | Presentation and Feedback | Showcase projects with peer reviews. | Develop confidence in coding. |

## Grade 4, 5 and 6 (with slight complexities based on classes)

|  |  |  |  |
| --- | --- | --- | --- |
| Week | Topic | Activities | Objectives |
| 1 | Review of First-Term Topics | Recap key concepts. | Reinforce knowledge. |
| 2 | Advanced Scratch | Work on interactive Scratch projects. | Improve problem-solving skills. |
| 3 | Scratch Game Development | Create and enhance games in Scratch. | Develop creativity and logic. |
| 4 | Advanced Scratch Techniques | Use clones, messages, and effects. | Strengthen Scratch programming skills. |
| 5 | Debugging in Scratch | Identify and fix Scratch program bugs. | Improve troubleshooting abilities. |
| 6 | Introduction to Web Development | Learn HTML basics. | Build web development skills. |
| 7 | Styling with CSS | Use CSS for page styling. | Understand how to structure web pages. |
| 8 | Integrating HTML & CSS | Create a simple webpage project. | Develop complete web pages. |
| 9 | Debugging Web Projects | Identify and fix web design errors. | Strengthen problem-solving abilities. |
| 10 | Final Project Planning | Plan an HTML/CSS-based project. | Apply all learned concepts. |
| 11 | Project Development | Work on web development projects. | Enhance coding independence. |
| 12 | Final Project Refinement | Debug and refine projects. | Improve project quality. |
| 13 | Presentation and Feedback | Showcase projects with peer reviews. | Develop confidence in coding. |

**Grade 5**

(Same structure as Grade 4, but with increased complexity and expectations in web development.)

**Grade 6**

(Similar structure, with additional emphasis on refining projects and independent learning.)