

SCO INTERNATIONAL OLYMPIAD

CLASS 3 CODING SYLLABUS

A practical syllabus guide for schools, teachers, parents, and students

Designed for Class 3 coding pathways and aligned with SCO's platform flow for guided preparation, practice, reporting, and future-ready digital learning.

- age-fit coding guidance for Class 3 / primary-level learners globally
- chapter-wise pathways across Fundamentals, Logic, Sprites, Practice Studio, and Teacher Guidance
- preparation roadmap, classroom ideas, and future-benefit framing for early digital confidence

Maths	Fundamentals	Logic	Sprites	Practice Studio
AI	Algorithms	Debugging	Puzzles	Teacher Guide

Official Syllabus Overview

SCO International Coding Olympiad - Class 3

Purpose of this syllabus

This syllabus introduces coding as a way of thinking: giving clear instructions, spotting patterns, solving puzzles, testing ideas, and creating simple interactive projects. It is designed for Class 3 learners and supports students, teachers, parents, and schools with clear chapter goals and practice guidance.

Chapter Snapshot

Chapter	Chapter Name	Learning Focus
1	Fundamentals of Coding	Instructions, sequence, algorithm, program, debugging, basic digital safety
2	Logic and Reasoning, solve puzzles using coding thinking	Patterns, step order, conditions, true/false thinking, puzzle reasoning
3	Simple Game Creation using sprites	Sprites, stage, events, motion, simple controls, score and feedback
4	Olympiad Practice Studio coding and learning for kids	MCQ practice, reasoning drills, scenario questions, project reflection
5	Teachers guide for Coding Olympiad for kids	Classroom flow, assessment guidance, inclusive practice, project-based support

Overall Learning Outcomes

- Explain that computers follow exact instructions written by people.
- Break a simple task into ordered steps and call it an algorithm.
- Recognize basic coding ideas: sequence, loop, condition, event, variable, output, and debugging.
- Use logic to solve simple puzzles and identify missing or incorrect steps.
- Understand sprites, stage, movement, keyboard input, collision, score, and simple game rules.
- Practice safe, responsible, and creative use of digital tools.

Competency Map

Competency	What Students Practice
Computational Thinking	Sequence, decomposition, pattern recognition, algorithmic steps
Coding Concepts	Events, loops, conditionals, variables, output, debugging
Creative Building	Sprites, costumes, stage, movement, animation, simple game rules
Reasoning Skills	Odd-one-out, direction logic, simple math logic, cause-effect questions
Olympiad Readiness	Question interpretation, answer elimination, accuracy, explanation writing

Coding begins with clear instructions. In this chapter, students learn that a computer cannot guess what a person wants; it follows the exact steps given to it. Learners connect everyday routines such as brushing teeth, packing a school bag, or drawing a shape with the idea of sequence and algorithm.

Focus Area	Details
Core concepts	Computer as an instruction follower Sequence and step order Algorithm and program Input and output Basic debugging and safe use
Learning outcomes	Define coding as writing instructions for a computer. Explain the difference between an instruction, an algorithm, and a program. Arrange simple steps in the correct order. Identify basic errors and understand why debugging is needed.
Practice ideas	Write steps to make a sandwich or open a book. Act as a robot and follow classmates instructions exactly. Find the missing or wrong step in a short instruction list.
Assessment focus	Students should be able to identify what coding means, choose the correct order of steps, and explain why a computer needs clear instructions.

CHAPTER 2**Logic and Reasoning, solve puzzles using coding thinking**

Coding requires students to think carefully before giving instructions. This chapter builds logical thinking through patterns, direction movement, true/false statements, and puzzle-style reasoning. Students learn to check whether a rule works and to predict the next step.

Focus Area	Details
Core concepts	Patterns and sequences True or false logic If-this-then-that thinking Robot movement and direction Odd-one-out and missing-step puzzles
Learning outcomes	Recognize number, shape, color, and movement patterns. Use true/false reasoning to test simple statements. Solve step-by-step movement and direction puzzles. Explain why an answer follows a rule.
Practice ideas	Complete color and number patterns. Trace a robot path on grid paper. Sort statements into always true, sometimes true, and false. Solve maze-style or direction-based puzzles.
Assessment focus	Assessment should check whether students can apply a rule, not only calculate an answer. Explanations should be simple and based on the pattern or condition.

CHAPTER 3
Simple Game Creation using sprites

This chapter introduces children to interactive creation. Students learn that a sprite is a character or object in a project and that sprites can move, speak, hide, show, change costume, and respond to events. Learners connect coding blocks with visible outcomes in simple games.

Focus Area	Details
Core concepts	Sprite and stage Events such as green flag or key press Motion and direction Show and hide Simple collision or touching rules Score and feedback
Learning outcomes	Identify sprites, stage, costumes, and events in a block-based project. Use simple motion ideas such as move, turn, jump, and glide. Explain how keyboard input can control a sprite. Understand simple game rules such as reaching a target or avoiding a wall.
Practice ideas	Plan a maze game with start and finish points. Make a sprite move using arrow-key logic. Create a simple collect-the-object game idea. Explain what should happen when a sprite touches a wall or target.
Assessment focus	Students should be able to connect a game action to the coding idea behind it, such as using an event for a key press or a condition for touching an object.

Olympiad practice helps students move from knowing a concept to applying it in new situations. This chapter develops exam readiness through carefully designed MCQs, scenario-based questions, and logic challenges that connect coding ideas with real-life problem solving.

Focus Area	Details
Core concepts	MCQ reading strategy Scenario-based coding questions Pseudocode tracing Error spotting Revision through mini challenges
Learning outcomes	Read a question carefully and identify the coding idea being tested. Use elimination to remove incorrect options. Trace simple pseudocode or block logic mentally. Write a short reason for the chosen answer.
Practice ideas	Solve 10-minute mixed practice sets. Explain one answer orally after each practice set. Create a small question bank from classroom examples. Review mistakes as debugging practice.
Assessment focus	Practice should measure accuracy, reasoning, concept clarity, and ability to learn from mistakes. Students should be encouraged to explain why the other options are not correct.

CHAPTER 5
Teachers guide for Coding Olympiad for kids

This chapter supports teachers and schools in delivering coding without making it only a screen-based activity. Many Class 3 coding ideas can be taught through unplugged tasks, movement games, classroom discussion, paper mazes, and short Scratch-style demonstrations.

Focus Area	Details
Core concepts	<ul style="list-style-type: none"> Unplugged coding activities Teacher-led block demonstrations Classroom questioning Peer explanation Rubrics and readiness checks
Learning outcomes	<ul style="list-style-type: none"> Plan a balanced coding lesson using demonstration, activity, discussion, and reflection. Use age-appropriate vocabulary for algorithms, loops, conditions, sprites, and debugging. Assess students through reasoning, not memorization alone. Support beginners and advanced learners through differentiated tasks.
Practice ideas	<ul style="list-style-type: none"> Use a human-robot activity to teach sequence. Use colored cards to teach conditionals. Use grid-paper mazes to teach movement logic. Use project showcase time for students to explain their choices.
Assessment focus	<ul style="list-style-type: none"> Schools should evaluate students through concept clarity, logical explanation, collaboration, safe digital behavior, and readiness for Olympiad-style reasoning.

Suggested Preparation Roadmap

Timeline	Preparation Focus
Week 1-2	Learn basic instructions, sequence, and algorithm examples from daily life.
Week 3-4	Practice patterns, robot directions, true/false logic, and simple if-this-then-that reasoning.
Week 5-6	Explore sprite, stage, event, motion, hide/show, and simple game rules.
Week 7-8	Solve Olympiad-style mixed practice sets and explain answers.
Final Review	Revise keywords, common mistakes, debugging examples, and game-based scenarios.

Readiness Indicators

- Can explain coding as giving exact instructions to a computer.
- Can arrange a familiar task into clear steps.
- Can identify a loop, condition, event, and variable in a simple example.
- Can predict what a sprite should do after a key press or green flag event.
- Can solve direction, pattern, and logic questions without rushing.
- Can describe a mistake as a bug and suggest how to fix it.

Quick Revision Keywords

Keyword	Simple Meaning
Algorithm	Step-by-step instructions to solve a problem.
Program	A set of coded instructions that a computer can follow.
Loop	A coding idea used to repeat instructions.
Condition	A rule that decides whether an action should happen.
Event	Something that starts an action, such as a green flag or key press.
Sprite	A character or object used in a Scratch-style project.
Debugging	Finding and fixing mistakes in instructions or code.
Variable	A place to store information such as score or name.

Student, teacher, and school focus

Students should learn coding as a practical thinking skill. Teachers can combine unplugged activities with simple block-based demonstrations. Schools can use the syllabus to build confidence, creativity, logical reasoning, and early digital literacy in a safe and age-appropriate way.