School Connect Online

SCO INTERNATIONAL OLYMPIAD

CLASS 4 CODING SYLLABUS

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

Designed for Class 4 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 4 learners globally
- chapter-wise pathways across Fundamentals, Logic, Game Creation, Practice Studio, and Coding Scenarios
- preparation roadmap, implementation ideas, and future-benefit framing for digital creativity and computational thinking

Fundamentals	Logic	Game	Creation	Practice Studio
Algorithms	Debugging	Variables	Scratch	Safety

School Connect Online

Official Syllabus Overview

SCO International Coding Olympiad - Class 4

This syllabus introduces coding as a practical problem-solving skill. Students learn how instructions, logic, conditions, variables, debugging, and simple game design work together to create interactive digital projects. The structure supports students, teachers, and schools with clear learning outcomes, practice focus, classroom activities, and revision indicators.

Learning Positioning

The syllabus is designed for primary-level learners who are ready to move from simple computer awareness into computational thinking, block-based programming, basic game logic, debugging habits, and responsible use of technology.

Chapter Snapshot

Chapter	Chapter Name	Core Focus	Student Output	Classroom Use
1	Fundamentals of Coding	instructions, algorithm, sequence, variables, input and output	write clear steps and explain simple code behavior	unplugged activities and block-code demos
2	Logic and Reasoning used in Coding	patterns, conditions, decisions, tracing and debugging	solve logic puzzles and predict program output	reasoning worksheets and flowchart tasks
3	Simple Game Creation	sprites, events, motion, scoring, collision and loops	design a small interactive game or animation	Scratch-style project practice
4	Olympiad Practice Studio and Latest Scenarios in Coding for Kids	real-world digital scenarios, safe coding habits, robotics and AI awareness	attempt scenario-based olympiad questions confidently	guided revision and challenge rounds

Quick Use Guide for Schools and Teachers

The syllabus can be used for weekly preparation, topic-wise practice, project-based classroom sessions, revision worksheets, and olympiad-style assessment. It keeps coding practical, age-fit, and connected to problem-solving rather than memorization.

School Connect Online

Class 4 Coding Learning Outcomes

What students should understand, practise, and demonstrate

By the end of this syllabus, students should be able to:

- Explain that a computer follows exact instructions written by humans.
- Convert a simple everyday task into step-by-step instructions or an algorithm.
- Recognize common coding ideas such as sequence, loop, condition, event, variable, and output.
- Trace a small block of code or pseudocode and predict what will happen.
- Use logical thinking to identify missing steps, wrong order, and simple bugs.
- Understand how game elements such as sprites, movement, score, levels, collision, and keyboard control work.
- Apply safe, responsible, and creative thinking while using digital tools.

Competency Map

Skill Area	Expected Behaviour	Practice Type	Readiness Evidence
Computational Thinking	break a task into smaller steps	daily-life algorithms, sequencing cards	explains the order of steps clearly
Programming Concepts	use loops, conditions and variables in simple cases	Scratch/block-code tasks	predicts program output correctly
Debugging	spot simple mistakes in code or logic	error-finding exercises	suggests a fix with reason
Game Design	connect events, sprites and score logic	mini-game challenges	builds or describes a working game flow
Responsible Technology	understand safe and ethical digital behaviour	scenario questions	chooses safe and respectful actions

School Connect Online

Chapter 1: Fundamentals of Coding

Chapter note, learning outcomes and practice direction

Chapter Note

This chapter builds the base of coding literacy. Students understand that code is a set of instructions, programs are written to solve problems, and computers follow commands exactly as given. The chapter connects everyday instructions with algorithms, sequencing, variables, input, output and debugging.

Learning Outcomes

- Define coding, program, algorithm, instruction, input and output in simple language.
- Arrange steps in the correct sequence to solve a task.
- Identify examples of variables, loops and conditions in beginner-friendly contexts.
- Recognize errors in simple instructions and suggest corrections.
- Understand that testing and debugging improve a program.

Classroom and Home Practice

- Write steps to brush teeth, pack a school bag or cross a maze safely.
- Use arrow instructions to move a character on a grid.
- Create a simple variable such as score, lives or timer.
- Find the missing or incorrect step in a sample algorithm.

Olympiad Assessment Focus

Questions may ask students to choose the correct instruction order, identify valid variables, understand loops, select the right flowchart symbol, and explain debugging in simple terms.

Quick Revision Keywords

algorithm, code, program, sequence, input, output, variable, loop, condition, bug, debug

School Connect Online

Chapter 2: Logic and Reasoning used in Coding

Chapter note, learning outcomes and practice direction

Chapter Note

This chapter develops the reasoning habits needed for coding. Students practise pattern recognition, conditional thinking, comparison, sequencing, true/false decisions, and step-by-step tracing. The aim is to help students think like young problem-solvers before writing longer programs.

Learning Outcomes

- Solve number, shape, direction and pattern-based coding logic questions.
- Use if-then thinking to decide what a program should do next.
- Trace simple program statements and predict the result.
- Apply AND, OR and NOT reasoning in age-appropriate situations.
- Recognize why a wrong condition can make a program behave incorrectly.

Classroom and Home Practice

- Complete pattern puzzles and explain the rule used.
- Trace a robot path using left, right, forward and repeat commands.
- Convert a classroom rule into an if-then statement.
- Compare two conditions and decide whether an action should happen.

Olympiad Assessment Focus

Questions may include output prediction, missing pattern, direction logic, condition-based choices, flowchart decisions and debugging of simple logic errors.

Quick Revision Keywords

logic, reasoning, pattern, condition, true, false, trace, compare, decision, operator

School Connect Online

Chapter 3: Simple Game Creation

Chapter note, learning outcomes and practice direction

Chapter Note

This chapter introduces game-building through block-based thinking. Students learn how characters move, react, score points, collide with objects, change appearance and respond to user actions. Game creation helps students see coding as creative problem-solving.

Learning Outcomes

- Identify sprites, backdrop, costumes, events, motion and control blocks.
- Explain how keyboard events can control a character.
- Use loops and conditions to repeat actions and create game rules.
- Understand score, lives, timer and level variables in a game.
- Describe how collision detection can trigger actions such as score changes or game over.

Classroom and Home Practice

- Plan a simple catch, chase or maze game on paper before coding.
- Design rules for movement using arrow keys.
- Add a score variable and describe when it changes.
- Test a game and list one bug and one improvement.

Olympiad Assessment Focus

Questions may ask students to choose Scratch-style blocks, identify game logic, select an event, understand score changes, explain sprite movement and find errors in a game script.

Quick Revision Keywords

sprite, event, backdrop, costume, movement, score, collision, broadcast, forever loop, game rule

School Connect Online

Chapter 4: Olympiad Practice Studio and Latest Scenarios in Coding for Kids

Chapter note, learning outcomes and practice direction

Chapter Note

This chapter prepares students for scenario-based coding questions. It connects coding with robotics, games, digital safety, AI awareness, sensors, smart devices and responsible technology use. Students practise interpreting real-world situations and choosing the best coding or logic-based response.

Learning Outcomes

- Apply coding concepts to real-life and olympiad-style scenarios.
- Understand how sensors, robots, games and smart devices follow instructions.
- Choose safe and responsible actions in digital environments.
- Recognize simple uses of AI and automation without confusing them with human thinking.
- Attempt mixed revision questions involving algorithms, variables, logic and debugging.

Classroom and Home Practice

- Discuss how traffic lights, elevators, games and robots follow instructions.
- Solve short scenario questions on safe technology use.
- Create a challenge card: problem, input, rule, output and expected result.
- Practise mixed olympiad rounds with explanation, not guessing.

Olympiad Assessment Focus

Questions may include latest technology scenarios, coding ethics, safe digital behaviour, robotics path logic, AI awareness, smart-device examples and applied debugging problems.

Quick Revision Keywords

scenario, robotics, sensor, automation, AI awareness, digital safety, smart device, challenge, revision

School Connect Online

Preparation Roadmap

A practical route for students, teachers and schools

Stage	Student Focus	Teacher Focus	School Support
Week 1-2	learn instructions, sequence and algorithm basics	use unplugged games and step cards	provide access to lab/projector or printed worksheets
Week 3-4	practise loops, variables, if-then logic and output tracing	run mini debugging tasks	organize topic-wise practice sessions
Week 5-6	build simple animation or game flow	guide sprite, event, score and collision logic	display student projects or demos
Week 7-8	attempt mixed olympiad questions with explanations	review errors and reasoning steps	conduct short mock rounds and feedback

Recommended Practice Method

Students should first explain the logic in words, then draw the steps, and only then choose blocks or code. This helps them avoid guessing and builds real computational thinking.

Assessment Blueprint

Area	Indicative Weight	Sample Question Style	Skill Tested
Fundamentals of Coding	30%	What does a loop do? Which is a valid variable?	concept clarity
Logic and Reasoning	25%	What is the next step/output? Which condition is true?	reasoning and tracing
Simple Game Creation	25%	Which block changes score? Which event starts a game?	project understanding
Latest Scenarios and Practice Studio	20%	Which solution is safest or most logical in a real coding situation?	applied thinking

School Connect Online

Student Readiness Checklist

A quick glance before revision or mock practice

Ready Skill	Student Check	Teacher Observation
Can explain what coding means and why computers need clear instructions.	Yes / Almost / Needs Practice	Can explain with example
Can arrange steps of an algorithm in the correct order.	Yes / Almost / Needs Practice	Can explain with example
Can identify a loop, condition, variable, event and output in a simple example.	Yes / Almost / Needs Practice	Can explain with example
Can trace short pseudocode or block-code logic and predict the output.	Yes / Almost / Needs Practice	Can explain with example
Can explain how a sprite moves, scores points, changes costume or reacts to keys.	Yes / Almost / Needs Practice	Can explain with example
Can identify a bug and suggest a simple fix.	Yes / Almost / Needs Practice	Can explain with example
Can solve scenario questions using safe, responsible and logical thinking.	Yes / Almost / Needs Practice	Can explain with example
Can attempt mixed olympiad questions with explanations, not just by guessing.	Yes / Almost / Needs Practice	Can explain with example

Guidance for Schools and Teachers

- Use short demonstrations, hands-on tasks and discussion before formal testing.
- Encourage students to explain the logic behind each answer.
- Make debugging a positive habit by treating mistakes as clues for improvement.
- Use block-based coding projects to connect concepts with creativity.
- Include safe technology and responsible digital behaviour in every practice cycle.

Final Learning Note

Coding for Class 4 should build confidence, curiosity and structured thinking. The strongest students are not only those who remember terms, but those who can read a problem, break it into steps, test a solution, identify errors and explain their reasoning clearly.