School Connect Online

SCO INTERNATIONAL CODING OLYMPIAD

CLASS 7 OFFICIAL SYLLABUS

SCO International Coding Olympiad | Teacher, school and student reference

Designed as a structured Class 7 coding pathway that connects programming syntax, structured data, visual coding, game logic, inquiry and revision.

- chapter-wise small notes and measurable learning outcomes
- age-appropriate progression from basics to integrated problem solving
- aligned with international K-12 computer science and computational-thinking expectations

Programming	C Basics	XML	Turtle	Debugging
Algorithms	Data	Events	Projects	Review

School Connect Online

Official Syllabus - Class 7

The SCO International Coding Olympiad for Class 7 introduces students to programming through practical reasoning, debugging, structured data, visual graphics, and game-based computational thinking. The syllabus is written for global school use and supports teachers, parents, and students with clear chapter notes, expected learning outcomes, and assessment-ready competency descriptors.

Exam Details

Detail	Description
Exam Name	SCO International Coding Olympiad
Class / Grade	Class 7
Duration	60 minutes
Type of Exam	Objective type multiple-choice questions
Suggested Question Count	50 questions
Primary Skill Focus	Computational thinking, programming logic, debugging, structured data, Turtle graphics, game-development reasoning
Eligibility	Class 7 students

Chapter-wise Syllabus with Small Notes and Learning Outcomes

Chapter No.	Chapter Name	Small Notes for Learning	Learning Outcomes
1	Programming Basics	Introduces C syntax, variables, data types, operators, input/output and simple control flow. Students learn that code is a precise set of instructions and that small syntax mistakes can change program behavior.	Identify correct syntax, trace simple C programs, explain variables and data types, and detect common beginner errors such as missing semicolons or incorrect scanf usage.
2	Advanced Coding Concepts	Extends basic programming into functions, arrays, pointers, recursion, memory safety and logical debugging. The focus is not just recall but explaining why code fails or succeeds.	Analyze code snippets, predict output, recognize undefined behavior, evaluate array bounds and pointer use, and apply safe debugging habits.
3	Game Development Basics	Connects coding with interactive projects using Turtle graphics, simple game loops, keyboard events and visual feedback. Students learn how movement, events and updates produce interactive behavior.	Describe the purpose of a game loop, use Turtle commands conceptually, reason about event-driven programming, and identify how input changes game state.
4	Olympiad Studio: Revision, Inquiry and Latest Scenarios	Integrates C, XML and Turtle into practical scenarios such as scoreboards, game settings, graphical output and code-review tasks. It	Solve scenario-based MCQs, connect XML data to coding projects, evaluate code for correctness and safety, and communicate reasoning

		prepares students for higher-order Olympiad questions.	clearly through answer explanations.

Detailed Topic Scope

Domain	Topics Included	Class 7 Pedagogical Focus
C Programming Language Basics	printf, scanf, variables, char/int/float, arrays, functions, pointers, loops, conditional statements, simple debugging	Students should trace short code segments, understand memory/address basics at an introductory level, and avoid unsafe or undefined operations.
Basics of XML	Elements, root element, attributes, self-closing tags, proper nesting, comments, escaping special characters, XML for game or app data	Students should understand XML as structured data, not as a programming language, and identify well-formed versus malformed snippets.
Turtle Programming in Python	forward, left/right, circle, penup/pendown, goto, color/fill, keyboard events, simple shapes and movement	Students should connect geometric reasoning with simple programming commands and explain how repeated instructions create patterns.
Game Development Basics	Game loop, input handling, update/render cycle, score variables, collision or movement logic, external configuration files	Students should reason about interaction, timing, state change and the separation of data from program logic.
Debugging and Safety	Out-of-bounds array access, missing address operators, optional/nullable or uninitialized values, incorrect conditions, SQL/security awareness at introductory level	Students should recognize typical code defects and select the safest correction rather than guessing the output only.

Global Standard Alignment

Reference Framework	Class 7 Alignment Use
CSTA K-12 Computer Science Standards	Supports learning objectives around computing systems, data and analysis, algorithms and programming, and impacts of computing.
K-12 Computer Science Framework	Supports computational-thinking practices such as recognizing computational problems, developing abstractions, creating artifacts, testing/refining, and communicating about computing.
Python official Turtle documentation	Supports visual programming and geometric drawing activities appropriate for middle-school learners.
W3C XML 1.0 Recommendation	Supports structured-data literacy through well-formed elements, attributes, nesting and data exchange.
OWASP secure coding guidance	Introduces safe input and query-handling habits in age-appropriate debugging questions.

Teacher Notes and Assessment Rubric

Skill Band	Evidence of Learning	Teacher Action
Foundational	Student can identify syntax, tags and basic Turtle commands.	Use tracing tasks and short correction exercises.
Proficient	Student can predict output and explain why an option is correct.	Add mixed code-and-concept MCQs and require one-line reasoning.
Advanced	Student can diagnose ambiguous or unsafe code and select the safest correction.	Use case studies, debugging challenges and project-style questions.
Olympiad Ready	Student connects programming, XML data and Turtle/game logic in unfamiliar scenarios.	Use timed mixed-section practice and post-test explanation review.

Preparation Roadmap

Weeks 1-2: Revise C basics and XML structure. Weeks 3-4: Practice Turtle graphics, shape logic and event thinking. Weeks 5-6: Solve debugging and scenario-based questions. Final week: Complete a timed sample paper, review explanations, and maintain an error notebook covering syntax, logic, data and safety mistakes.

Recommended Question Design for Schools

A balanced Class 7 paper should include direct concept questions, output-prediction questions, error-detection questions, structured-data/XML questions, Turtle/game-logic questions and higher-order integrated scenarios. The Achievers Section should reward reasoning and careful code reading rather than memorization.