School Connect Online

SCO INTERNATIONAL CODING OLYMPIAD

CLASS 6 SYLLABUS

Official Syllabus 2026-27 | Programming, Web Basics and Game Development

A globally aligned, student-friendly Class 6 coding pathway for schools, teachers and learners.

- Built around computational thinking, programming foundations, web literacy, safe digital habits and creative application.
- Includes chapter-wise notes, learning outcomes, assessment blueprint and classroom implementation guidance.
- Useful for Olympiad preparation, school enrichment, clubs, coding labs and parent-guided practice.

Intro Programming	Python	Java	HTML	Game Basics	
Algorithms	Web Basics	Debugging	Data Thinking	SCO	

School Connect Online

SCO International Coding Olympiad - Class 6

The Class 6 SCO International Coding Olympiad introduces learners to programming concepts, basic algorithms, web page structure, styling, simple game logic and practical problem solving. The syllabus is designed for early middle-school students who are ready to move from computer awareness to structured computational thinking and small creative projects.

Exam Snapshot

Field	Official Detail
Exam Name	SCO International Coding Olympiad
Class / Eligibility	Class 6
Duration	60 minutes
Type of Exam	Objective Type MCQ
Number of Questions	45 questions
Core Sections	Intro to Programming; Basic Coding Applications; Game Development Basics; Achievers Section
Syllabus Topics	Computer science foundations; Java and Python basics; Python vs Java; HTML; CSS; basic algorithms; debugging; creative game/application thinking

Pedagogical Purpose

- Build confidence with programming vocabulary and simple syntax without overloading students with professional-level tooling.
- Develop computational thinking through sequencing, loops, variables, conditions, debugging and simple algorithms.
- Connect coding to visible outcomes such as webpages, game objects, scoring, movement and user interaction.
- Prepare students for higher grades where they will meet data science, application development, AI and secure web programming.

Chapter-wise Syllabus with Small Notes and Learning Outcomes

No.	Chapter	Small Notes for Learning	Learning Outcomes
1	Intro to Programming	Introduces computer science, instructions, sequencing, variables, inputs, outputs and the idea that programs solve problems through clear steps.	Identify basic programming terms; trace simple instructions; explain how computers follow step-by-step commands.
2	Basic Coding Applications	Connects programming concepts to small real-world tasks such as calculators, simple forms, text output, lists and decision-based actions.	Apply variables, conditions and loops in simple MCQ scenarios; predict outputs; choose correct syntax for basic tasks.
3	Game Development Basics	Builds understanding of sprites, events, movement, scoring, collision detection, game loops and repeated actions through beginner-friendly contexts.	Recognize game objects and events; explain collision and score logic; select steps for simple game behavior.
4	Difference between Python and Java	Compares indentation and braces, dynamic and static typing, interpretation and compilation, syntax style and use cases.	Differentiate Python and Java features; identify correct variable and function syntax; reason about simple code snippets.

No.	Chapter	Small Notes for Learning	Learning Outcomes
5	What is HTML (HyperText Markup Language)?	Explains HTML as the structure of a webpage, including headings, paragraphs, links, images, lists, forms and semantic elements.	Choose correct tags; identify valid nesting; understand how HTML gives meaning and structure to webpage content.
6	Introduction to CSS	Introduces Cascading Style Sheets for color, font size, background, selectors, classes, IDs, box model basics and visual presentation.	Select correct CSS properties; understand selector specificity at a basic level; connect style rules to webpage output.

Recommended Assessment Blueprint

Section	Suggested Questions	Focus
Intro to Programming	10-12	Computer science vocabulary, Python/Java basics, syntax recognition, variables and outputs
Basic Coding Applications	12-14	HTML/CSS basics, functions, loops, conditions, lists, simple algorithms and debugging
Game Development Basics	8-10	Events, sprites, score logic, collision detection, repeated actions and game loops
Achievers Section	8-10	Higher-order output prediction, error analysis, short case studies and reasoning-based coding questions

Learning Progression for Class 6

Remember and Understand

Students recognize terms, symbols, tags, syntax and basic programming rules.

Apply

Students select correct code, predict simple output and identify how a webpage or game element behaves.

Analyze

Students find errors in small snippets, compare languages and explain why one option is more correct than another.

Create

Students are encouraged to build a simple web page, small Python program, or game logic flow as enrichment practice.

Teacher and School Implementation Notes

- Use short code-tracing activities before MCQ practice so students learn to read code line by line.
- For HTML/CSS, let students view simple pages in a browser so tags and styles become visible and meaningful.
- For game-development topics, connect concepts to sprites, movement, score changes, winning conditions and collision examples.
- Encourage students to explain why incorrect options are wrong; this improves debugging and exam accuracy.
- Use the Achievers Section for enrichment, coding club discussions and differentiated practice for high-performing learners.

Global Standard Alignment Snapshot

Reference Area	How this Class 6 syllabus aligns
K-12 Computer Science Framework	Uses computational-thinking practices such as problem decomposition, algorithms, abstraction, testing and communication.
CSTA-style Grade 6-8 Progression	Introduces algorithms and programming, data and analysis, computing systems, impacts of computing and web concepts at an age-appropriate level.
AI4K12 Readiness	Builds early readiness for AI learning through data, pattern recognition, logic, responsible use and computational thinking.
MDN Web Literacy	HTML and CSS topics follow modern web-literacy expectations: structure, semantics, styling, links, images, forms and selectors.
OWASP Secure Coding Awareness	Basic SQL/web-security questions introduce safe habits such as using prepared statements rather than direct string concatenation.

Preparation Roadmap for Students

- Week 1: Revise basic computer science terms, algorithms, Python statements and Java syntax.
- Week 2: Practice HTML tags, CSS properties, selectors, colors, fonts, links, images and simple forms.
- Week 3: Study loops, conditions, functions, lists, debugging and output prediction.
- Week 4: Practice game-development concepts, reasoning questions, mini-project scenarios and Achievers-level MCQs.

Reference Snapshot

This syllabus is aligned with international K-12 computer science principles and beginner web-development practice. Teachers may support students with browser-based HTML/CSS experiments, simple Python exercises, Java syntax comparisons, Scratch-style game logic and safe web-development discussions.