School Connect Online

SCO INTERNATIONAL OLYMPIAD

GRADE 9 CODING OLYMPIAD

Official syllabus for students, teachers, schools, and parents

Designed from Class 9 coding pathways and aligned with global computer science readiness

- grade-fit coding guidance for Class 9 / secondary-level learners globally
- programming concepts, application development, advanced Python, data analytics, Swift, Objective-C, PHP, SQL, and project-based coding
- practice roadmap, assessment readiness, secure coding awareness, and future-ready computational growth

Python	Algorithms	Swift	Objective-C	SQL
Data Science	Web Apps	Debugging	AI Basics	Security

SCO International Coding Olympiad - Class 9 Official Syllabus

Class 9 students benefit from the SCO International Coding Olympiad by advancing programming knowledge through control structures, algorithms, application development and practical coding scenarios. The syllabus builds computational thinking, debugging ability, data literacy, secure coding awareness and project-based confidence for future technology courses and competitions.

Detail	Description
Exam Name	SCO International Coding Olympiad
Class / Grade	Class 9
Duration	60 minutes
Type of Exam	Objective Type / MCQ-based reasoning and code-analysis assessment
Number of Questions	50 questions
Eligibility	Class 9 students
Core Focus	Functions, algorithms, application development basics, Swift, Objective-C, PHP, SQL, advanced Python, data science and basic statistics with Python
Assessment Style	Programming concepts, application development, coding projects, debugging and achievers-level scenario analysis

Global Curriculum Alignment

Alignment Area	Class 9 Learning Emphasis
Computational Thinking	Decompose problems, recognize patterns, use abstraction and design step-by-step algorithms.
Algorithms and Programming	Read, trace, debug and reason about Python, C/Objective-C style logic, Swift, PHP and SQL snippets.
Data and Analysis	Use introductory statistics, Pandas-style operations, grouping, filtering and simple data-science workflows.
Secure and Responsible Coding	Recognize SQL injection risks, safe user-input handling, authentication basics and error-aware development.
Project-Based Application	Connect code to real-world web apps, mobile apps, data dashboards and mini AI/data scenarios.

Chapter-wise Syllabus with Notes and Learning Outcomes

No.	Chapter	Small Notes for Learning	Learning Outcomes
1	Programming Concepts	Introduces functions, variables, loops, conditionals, recursion, lambda expressions and basic complexity reasoning.	Trace code, predict output, identify syntax/logic errors and choose efficient basic programming constructs.

2	Application Development	Connects programming logic to web and app development using HTTP, sessions, APIs, frontend-backend flow and database interaction.	Explain how applications retrieve, process and present data through safe and structured client-server workflows.
3	Advanced Python	Covers list comprehensions, decorators, generators, error handling, file/data handling and common Python mistakes.	Use Pythonic constructs responsibly, debug common errors, and reason about memory-friendly processing.
4	Coding Projects	Applies coding concepts through student-management systems, recommendation tasks, authentication flows and data-processing mini-projects.	Select suitable code approaches for real scenarios and justify implementation choices using logic, safety and scalability.
5	Data & Analytics Extensions	Builds basic statistics with Python, Pandas operations, data cleaning, grouping, aggregation and simple machine-learning concepts.	Analyze small datasets, compute summary measures, and choose correct tools for classification, recommendation and prediction contexts.
6	Olympiad Project Studio (Basic to Advanced)	Encourages structured thinking from simple programs to project workflows with inputs, processing, outputs, testing and improvement.	Design a small project workflow, identify edge cases, test outputs and refine the solution.
7	Theory Enquiries, Contemporary Scenarios and Assessments for Advanced Coding	Uses contemporary examples such as secure login, data dashboards, e-commerce logic, APIs and AI-assisted classification.	Apply coding knowledge to new scenarios, evaluate options critically and select the most reliable solution.
8	Teacher Notes, Rubrics, Differentiation and Reference Snapshot for Advanced Coding Learning	Provides educators with classroom-useful progression, assessment lenses and support for mixed-ability learners.	Use rubrics to evaluate accuracy, logic, debugging, explanation quality and real-world application readiness.

Teacher and School Implementation Guidance

Suggested learning pathway

Start with code tracing and output prediction, then move to debugging and scenario-based reasoning.

Use short practical demonstrations in Python, SQL and web-development contexts before attempting MCQs.

Encourage students to explain why incorrect options fail, especially in security and debugging questions.

Use the achievers section for advanced learners who are ready for multi-step reasoning and real-world application design.

Assessment Blueprint

Detail	Description
Programming Concepts	Approx. 30-35% - syntax, logic, control structures, functions, basic algorithms and code tracing.
Application Development	Approx. 20-25% - HTTP, sessions, APIs, SQL, PHP and web/mobile development reasoning.
Data & Analytics	Approx. 15-20% - Python data handling, Pandas, statistics and simple ML/data scenarios.
Coding Projects	Approx. 15-20% - applied problem-solving, architecture choices and mini-project decisions.
Achievers Section	Approx. 10-15% - higher-order debugging, security reasoning and multi-step coding scenarios.

Global Reference Snapshot

• CSTA K-12 Computer Science Standards - Computing Systems, Data and Analysis, Algorithms and Programming, and Impacts of Computing.

• K-12 Computer Science Framework - computational problem solving, abstraction, algorithms, programming, data and impacts.

• AI4K12 initiative - foundational AI literacy and responsible understanding of machine-learning systems.

• OWASP secure-coding guidance - prepared statements, input handling and SQL-injection prevention.