School Connect Online

SCO INTERNATIONAL CODING OLYMPIAD

CLASS 12 OFFICIAL QUESTION PAPER

Question Paper Set A • 2026-27 • 50 objective questions • 60 minutes

Designed for Class 12 coding readiness, advanced problem solving, and global computer-science preparation.

- advanced data structures: arrays, linked lists, trees, graphs, traversal and complexity
- algorithmic thinking with Python, C, Swift, PHP, SQL, APIs, web applications and debugging
- AI/ML and data-science basics with secure coding and real-world application reasoning

Python	C	Swift	PHP	SQL
Data Structures	Algorithms	AI / ML	Web Apps	Security

School Connect Online

Candidate Information

Exam Name	SCO International Coding Olympiad
Class / Grade	Class 12
Question Paper Set	Set A
Academic Year	2026-27
Duration	60 minutes
Total Questions	50
Question Type	Objective multiple-choice questions
Calculator	Not allowed unless specifically announced by SCO

Guidelines for the Candidate

• Read every question carefully and select only one correct option.

• All questions are compulsory. There is no negative marking in this practice-ready version unless the official event notification states otherwise.

• Use the question paper for reasoning and rough work only; mark responses on the answer sheet or online test interface as instructed.

• Code snippets should be interpreted exactly as printed. Assume standard language behavior unless the question states otherwise.

• For security, AI and data-science questions, choose the option that follows safe engineering practice and validated reasoning, not only surface-level syntax.

School Connect Online

Question Paper

Section A: Programming Fundamentals, Data Structures and Complexity

Q1. What does the following Python code output?

```
arr = [1, 2, 3, 4, 5]
print(arr[2] + arr[-1])
```

A. 5
B. 6
C. 7
D. 8

Q2. What does this loop print?

```
for i in range(3):
    print(i * 2, end=" ")
```

A. 0 1 2
B. 0 2 4
C. 1 2 3
D. 2 4 6

Q3. In a singly linked list, what does each node normally contain?
A. Data only
B. Data and a link/reference to the next node
C. Data and links to both next and previous nodes
D. Data and index number only

Q4. Which node in a tree has no parent?
A. Leaf node
B. Internal node
C. Root node
D. Child node

Q5. In graph theory, what do we call vertices that are directly connected by an edge?
A. Siblings
B. Adjacent vertices
C. Parallel nodes
D. Terminal nodes

Q6. What does this code calculate?

```
numbers = [10, 20, 30, 40, 50]
average = sum(numbers) / len(numbers)
```

A. Median
B. Mode
C. Mean
D. Range

Q7. Which SQL keyword is used to retrieve data from a database table?
A. GET
B. FETCH
C. SELECT
D. RETRIEVE

Q8. What is the output of this function call?

```
def greet(name="Student"):
    return f"Hello, {name}!"
print(greet())
```

A. Hello, !
B. Hello, Student!
C. Error
D. Hello, name!

Q9. Which HTML tag is used to create a hyperlink?
A. <link>
B. <a>
C. <href>
D. <url>

Q10. Which type of machine learning uses labeled input-output examples to learn prediction patterns?
A. Unsupervised learning
B. Reinforcement learning
C. Supervised learning
D. Unlabeled clustering

Q11. What does this code output?

```
numbers = [5, 2, 8, 1]
numbers.sort()
print(numbers[1])
```

A. 1
B. 2
C. 5
D. 8

Q12. How many times does this loop execute?

```
count = 0
while count < 3:
    count += 1
```

A. 2 times
B. 3 times
C. 4 times
D. Infinite loop

Q13. What is the time complexity of inserting at the beginning of a singly linked list when the head pointer is available?
A. O(1)
B. O(n)
C. O(log n)
D. $O(n^2)$

Q14. Which traversal visits nodes in sorted ascending order for a Binary Search Tree?
A. Pre-order
B. In-order
C. Post-order
D. Level-order

Q15. What does an adjacency matrix usually use to represent whether an edge exists between two vertices?
A. Linked lists
B. Arrays of objects only
C. Boolean or numeric values such as 0 and 1
D. Hash tables only

Q16. What does this function calculate?

```
def calculate(values):
    return max(values) - min(values)
```

A. Mean
B. Median
C. Range
D. Variance

Q17. Which SQL clause filters rows before grouping or ordering?
A. GROUP BY
B. ORDER BY
C. WHERE
D. SELECT

Q18. What is the output?

```
text = "hello"
print(text.upper())
```

A. hello
B. HELLO
C. Hello
D. Error

Q19. Which CSS property changes the text color of an element?
A. text-color
B. font-color
C. color
D. text-style

Q20. What is the purpose of training data in supervised learning?
A. To test the final model only
B. To teach the model patterns from labeled examples
C. To deploy the model
D. To hide labels from the model

Section B: Debugging, Algorithms, Data Science and AI Basics

Q21. The following C code attempts to print all array elements. Identify the error.

```
int arr[5] = {1, 2, 3, 4, 5};
for (int i = 0; i <= 5; i++) {
    printf("%d ", arr[i]);
}
```

A. The array declaration should not specify a size.
B. The printf statement must be removed.
C. The loop condition should be i < 5 instead of i <= 5.
D. The array must be initialized with zeros only.

Q22. The code intends to insert a new node at the end of a singly linked list. What is the main logical error?

```
struct Node* last = *head_ref;
while (last != NULL) {
    last = last->next;
}
last->next = new_node;
```

A. new_node should not be allocated.
B. The loop should stop at while (last->next != NULL).
C. The function cannot accept a pointer to pointer.
D. The new node should point back to the head.

Q23. In the inorder traversal below, what is the error?

```
void inorderTraversal(struct TreeNode* root) {
    if (root == NULL) return;
    inorderTraversal(root->left);
    printf("%d ", root->data);
    inorderTraversal(root->left);
}
```

A. The base case should check both children.
B. The second recursive call should traverse root->right.
C. The print statement must come first.
D. Inorder traversal cannot use recursion.

Q24. The following DFS routine for an adjacency matrix contains a recursion error. Which fix is correct?

```
if (graph[vertex][i] == 1 && visited[i] == false) {
```

```
    DFS(graph, vertex, visited, n);
}
```

A. Make visited global.
B. Start the loop from i = 1.
C. Pass i to DFS instead of vertex.
D. Remove the recursive call.

Q25. The function below copies a string but contains a C string handling bug. What is missing?

```
void stringCopy(char dest[], char src[]) {
    int i = 0;
    while (src[i] != '\0') {
        dest[i] = src[i];
        i++;
    }
}
```

A. It should use strlen only.
B. It should use char* only.
C. It should add dest[i] = '\0' after the loop.
D. It cannot copy empty strings.

Q26. For an n × m matrix, which method allows constant-time rectangular submatrix-sum queries after preprocessing?
A. Dynamic allocation only
B. Prefix-sum matrix with inclusion-exclusion
C. Binary search on every row
D. Random sampling

Q27. A binary classifier is trained on data with 95% negative cases and 5% positive cases. It reports 96% accuracy, but recall for the positive class is 0. What is the best diagnosis and first corrective action?
A. Accuracy alone is sufficient; no issue exists.
B. The model likely predicts the majority class; use stratified splitting and class balancing or resampling.
C. Remove the minority class.
D. Replace all features with random noise.

Q28. An A/B test has p-value 0.000003 and conversion rates 12.3% and 14.5% with non-overlapping 95% confidence intervals. What is the correct conclusion?
A. Reject the null hypothesis; Variant B is significantly better, while checking independence and expected counts.
B. Fail to reject the null hypothesis.
C. Use a normality test on each individual conversion only.

D. Conclude there is no practical difference because the p-value is small.

Q29. For detecting a simple cycle of exact length k in a sparse graph when k is small, which approach is usually practical?

A. Depth-limited DFS/BFS from each vertex with path tracking

B. Always solve Hamiltonian cycle first

C. Sort all vertex names alphabetically

D. Use only a SQL GROUP BY query

Q30. A multi-task loss combines cross-entropy with a weighted MSE where outliers receive 5× weight and weights are not normalized. What is the main risk?

A. Regression outliers may dominate gradients and destabilize training.

B. Cross-entropy cannot be used with classification.

C. $\alpha + \beta$ must always be greater than 2.

D. MSE changes all labels to strings.

Section C: Practical Programming, Databases, Web and Security

Q31. What is the main bug in this Python async loop?

```
tasks = []
for client in clients[:5]:
    async def execute():
        return await client.query("SELECT * FROM data")
    tasks.append(asyncio.create_task(execute()))
```

A. SQL cannot run inside async code.

B. clients[:5] always returns an empty list.

C. The inner function captures the loop variable client late, so tasks may use the same final client.

D. create_task cannot accept coroutines.

Q32. What does this Swift code print?

```
var optionalString: String? = "Hello"
if let value = optionalString {
    print(value + " World!")
} else {
    print("No value")
}
```

A. Hello

B. Hello World!

C. No value

D. Runtime error

Q33. What is the output of this C code?

```
int arr[] = {10, 20, 30, 40, 50};
int *ptr = arr;
ptr += 3;
printf("%d", *ptr);
```

A. 10
B. 20
C. 40
D. 50

Q34. Which practice best prevents SQL injection when accepting a username from a web form?
A. Concatenate user input directly into SQL.
B. Use parameterized queries or prepared statements.
C. Store SQL queries in plain text files.
D. Print the query before execution.

Q35. What does this SQL query accomplish?

```
SELECT department, COUNT(*) AS EmployeeCount
FROM Employees
WHERE salary > 50000
GROUP BY department
HAVING COUNT(*) > 5
ORDER BY EmployeeCount DESC;
```

A. Retrieves all employees regardless of salary.
B. Counts high-salary employees by department, keeps departments with more than five such employees, and sorts by count.
C. Deletes departments with low salaries.
D. Updates every employee salary.

Q36. Which HTTP method is normally used to update an existing resource in a RESTful API?
A. GET
B. POST
C. PUT
D. TRACE

Q37. What is the output and memory behavior of this generator code?

```
def count_up_to(n):
    i = 1
    while i <= n:
        yield i
        i += 1
print(list(count_up_to(5)))
```

A. [1, 2, 3, 4, 5], but generators always use more memory than lists
B. [1, 2, 3, 4, 5], and generators can produce values lazily
C. [1, 2, 3, 4]
D. Error

Q38. What is the output of this decorator example?

```
def shout(func):
    def wrapper(*args, **kwargs):
        return func(*args, **kwargs).upper()
    return wrapper

@shout
def greet(name):
    return f"Hello, {name}"
print(greet("Alice"))
```

A. Hello, Alice
B. HELLO, ALICE
C. hello, alice
D. Error

Q39. What is the time complexity of BFS on a graph with V vertices and E edges when using an adjacency list?
A. O(V)
B. O(E)
C. O(V + E)
D. O(V × E)

Q40. For an imbalanced classification task, which metric is more informative than accuracy when the minority class matters?
A. File size
B. Screen resolution
C. Precision, recall and F1-score for the minority class
D. Number of CPU fans

Achievers Section: Advanced Problem Solving

Q41. Achievers: Why is Dijkstra's algorithm not suitable for graphs with negative edge weights?

A. It only works on trees.

B. Its greedy assumption can finalize a distance before a later negative edge would improve it.

C. It cannot use priority queues.

D. It requires every edge to have weight 1.

Q42. Achievers: A table is queried frequently by tenant_id and created_at range. Which index is most useful for this access pattern?

A. An index only on a random JSON field

B. A composite index on (tenant_id, created_at)

C. No index because range queries cannot use indexes

D. An index on created_at only is always best

Q43. Achievers: In JWT verification, which step must happen before trusting user id, role or expiry claims?

A. Decode the payload and immediately trust it.

B. Verify the signature and algorithm using a trusted key, then validate claims.

C. Store the token in local storage only.

D. Remove all expiration checks.

Q44. Achievers: What kind of bug is shown here?

```
int *p = malloc(sizeof(int));
free(p);
printf("%d", *p);
```

A. Safe pointer reuse

B. Memory allocation improvement

C. Use-after-free

D. Syntax error caused by malloc

Q45. Achievers: What is the purpose of memoization in recursive algorithms?

A. To deliberately repeat every subproblem

B. To cache results of repeated subproblems and reduce redundant computation

C. To remove base cases

D. To make all algorithms O(1)

Q46. Achievers: Several async tasks update the same shared state without ordering or locking. What is the most likely issue?

A. Compile-time HTML error

B. Race condition or nondeterministic state updates

C. SQL syntax highlighting
D. Automatic dead-code elimination

Q47. Achievers: Which data-science pipeline issue creates overly optimistic test performance?
A. Splitting the dataset before any preprocessing
B. Evaluating on a held-out test set
C. Fitting preprocessing steps on the full dataset before train-test split
D. Using labels only in the training set

Q48. Achievers: What is the time complexity of this code?

```
for i in range(n):
    j = 1
    while j < n:
        j *= 2
```

A. O(n)
B. O(log n)
C. O(n log n)
D. $O(n^2)$

Q49. Achievers: Which data structure is typically used to perform BFS traversal?
A. Queue
B. Recursion stack only
C. Binary heap only
D. Hash function only

Q50. Achievers: Which option gives the strongest secure coding response for a login API?
A. Log every password to simplify debugging.
B. Trust X-Forwarded-For from any client for rate limiting.
C. Disable token expiration for convenience.
D. Use prepared statements, never log passwords, validate tokens before trusting claims, and rate-limit using trusted proxy data.

Answer Key

Q.No.	Ans	Q.No.	Ans	Q.No.	Ans	Q.No.	Ans	Q.No.	Ans
1	D	11	B	21	C	31	C	41	B
2	B	12	B	22	B	32	B	42	B
3	B	13	A	23	B	33	C	43	B
4	C	14	B	24	C	34	B	44	C

Q.No.	Ans	Q.No.	Ans	Q.No.	Ans	Q.No.	Ans	Q.No.	Ans
5	B	15	C	25	C	35	B	45	B
6	C	16	C	26	B	36	C	46	B
7	C	17	C	27	B	37	B	47	C
8	B	18	B	28	A	38	B	48	C
9	B	19	C	29	A	39	C	49	A
10	C	20	B	30	A	40	C	50	D

Detailed Explanations

Q1. Answer: D. arr[2] is 3 and arr[-1] is the last element, 5. Their sum is 8.
Q2. Answer: B. range(3) generates 0, 1 and 2. Multiplying by 2 gives 0, 2 and 4.
Q3. Answer: B. A singly linked list node contains data and one reference/pointer to the next node.
Q4. Answer: C. The root is the topmost node of a tree and has no parent.
Q5. Answer: B. Two vertices sharing an edge are called adjacent vertices.
Q6. Answer: C. The sum divided by the number of values is the arithmetic mean.
Q7. Answer: C. SELECT is the standard SQL statement used to query data from tables.
Q8. Answer: B. No argument is supplied, so the default value Student is used.
Q9. Answer: B. The anchor tag <a>, usually with an href attribute, creates a hyperlink.
Q10. Answer: C. Supervised learning trains models using examples that include the correct label or target value.
Q11. Answer: B. After sorting, the list becomes [1, 2, 5, 8]. The item at index 1 is 2.
Q12. Answer: B. The loop executes for count values 0, 1 and 2, then stops when count becomes 3.
Q13. Answer: A. Only the new node and head pointer must be updated, so it is constant time.
Q14. Answer: B. In-order traversal of a BST visits left subtree, root and right subtree, producing sorted order.
Q15. Answer: C. An adjacency matrix stores edge presence using values such as 1 for an edge and 0 for no edge.
Q16. Answer: C. The difference between maximum and minimum values is the range.
Q17. Answer: C. WHERE restricts rows based on conditions before final grouping and ordering stages.
Q18. Answer: B. The upper() method converts all alphabetic characters to uppercase.
Q19. Answer: C. The CSS property for text color is color.
Q20. Answer: B. Training data is used to fit the model by learning patterns between inputs and labels.
Q21. Answer: C. Valid indices are 0 to 4. The condition i <= 5 tries to access arr[5], which is out of bounds.
Q22. Answer: B. The loop makes last NULL, so last->next dereferences a null pointer. It should stop at the last node.
Q23. Answer: B. Inorder traversal is left, root, right. The second recursive call should use the right child.
Q24. Answer: C. When an unvisited neighbor i is found, DFS must recurse on that neighbor, not the same current vertex.
Q25. Answer: C. A valid C string must end with a null terminator.
Q26. Answer: B. A 2D prefix-sum matrix preprocesses cumulative sums so any rectangle can be queried using inclusion-exclusion in O(1).
Q27. Answer: B. High accuracy can be misleading with imbalanced classes. Positive-class recall, precision, F1 and balancing strategies matter.
Q28. Answer: A. The p-value is far below 0.05 and the confidence intervals do not overlap, supporting a significant improvement for Variant B.
Q29. Answer: A. For small fixed k, depth-limited search explores paths up to length k and can be practical on sparse graphs.
Q30. Answer: A. Unnormalized loss scales can make one task dominate optimization, especially with weighted outliers.
Q31. Answer: C. Python closures capture variables by reference. Binding client as a default argument or defining a helper fixes it.
Q32. Answer: B. optionalString contains a value, so optional binding unwraps it safely.
Q33. Answer: C. ptr moves to arr[3], which stores 40.
Q34. Answer: B. Prepared statements separate code from data, preventing user input from being treated as SQL commands.
Q35. Answer: B. WHERE filters rows, GROUP BY groups departments, HAVING filters groups, and ORDER BY sorts the result.
Q36. Answer: C. PUT is commonly used to update or replace an existing resource.
Q37. Answer: B. The generator yields values one at a time. list() consumes all yielded values.
Q38. Answer: B. The decorator replaces greet with wrapper, which converts the returned string to uppercase.
Q39. Answer: C. BFS visits vertices and scans adjacency lists, giving O(V + E).

Q40. Answer: C. Minority-class precision, recall and F1 reveal whether the model detects the important rare class.
Q41. Answer: B. Dijkstra relies on non-negative weights so that once a node is chosen as minimum, its distance is final.
Q42. Answer: B. The composite index supports filtering by tenant and then scanning the relevant time range efficiently.
Q43. Answer: B. Claims are untrusted until the token signature and allowed algorithm are verified.
Q44. Answer: C. The pointer is dereferenced after the memory has been freed, which is a use-after-free bug.
Q45. Answer: B. Memoization stores results so repeated subproblems do not need to be recomputed.
Q46. Answer: B. Concurrent updates to shared state can occur in unpredictable order and produce inconsistent results.
Q47. Answer: C. Fitting preprocessing on the full dataset leaks information from the test data into training.
Q48. Answer: C. The outer loop runs n times and the inner loop doubles j, running O(log n) times.
Q49. Answer: A. BFS explores nodes level by level and normally uses a queue.
Q50. Answer: D. Secure login design combines input safety, secret protection, validated tokens and trustworthy rate limiting.