School Connect Online

SCO INTERNATIONAL CODING OLYMPIAD

CLASS 7 SAMPLE QUESTION PAPER

SCO International Coding Olympiad | Sample Paper Set B | With answers and explanations

Designed for Class 7 learners to build confident coding foundations through C programming, XML data thinking, Turtle graphics, and game-development logic.

- compact inline question numbers with no large question-number block
- clean question blocks with code, options, answer key, and explanations together
- global alignment with K-12 computer science, computational thinking, and safe coding practices

C Basics	XML	Turtle	Game Logic	Algorithms
Functions	Data	Events	Safety	Review

SCO International Coding Olympiad- Class 7 Sample Paper Set B | With answers and explanations

Field	Official Detail
Exam Name	SCO International Coding Olympiad
Class / Grade	Class 7
Duration	60 minutes
Type of Exam	Objective type multiple-choice questions
Number of Questions	50
Answer Format	One correct option per question
Calculator	Not allowed unless specifically announced by SCO
Website Use	PDF-ready student/teacher download

Candidate guidance: Read each question carefully, select only one answer, and use the explanation section for post-exam learning and teacher-supported review. The question number is intentionally shown as a compact inline label to keep the question, code, passage, options, answer, and explanation together in one academic block.

Section Blueprint

Section	Focus Area	Learning Emphasis
Section 1	Programming Basics	C syntax, variables, input/output, arrays, XML foundations and Turtle movement
Section 2	Advanced Coding Concepts	Debugging, functions, recursion, memory safety, data formats and event logic
Section 3	Game Development Basics	Game loops, Turtle graphics, XML configuration, user input and simple simulation
Achievers Section	Higher-order reasoning	Error analysis, real-world debugging and integrated coding scenarios

Section 1: Programming Basics

Q1. What is the correct syntax to print "Hello, World!" in C?

A) print("Hello, World!");
B) printf("Hello, World!");
C) System.out.println("Hello, World!");
D) cout << "Hello, World!";

Answer: B) printf("Hello, World!");

Explanation: In C language, we use the printf function from stdio.h to display output on the screen.

Q2. What is the correct data type for storing a single character in C?

A) string
B) char
C) character
D) text

Answer: B) char

Explanation: The char data type is used to store a single character in C, such as 'A' or 'B'.

Q3. What is the correct syntax to declare an integer variable in C?

A) int x;
B) x = int;
C) integer x;
D) var x: int;

Answer: A) int x;

Explanation: In C, an integer variable is declared using the int keyword, followed by the variable name.

Q4. What will be the output of the following C code?

```
#include <stdio.h>
int main() {
printf("%d", 5 + 3 * 2);
return 0;
}
```

A) 16
B) 11
C) 13
D) 10

Answer: B

Explanation: Multiplication has higher precedence than addition, so 3 * 2 = 6 and 5 + 6 = 11. Therefore the correct option is B.

Q5. What is the purpose of the return 0; statement in a C program?

A) To print output
B) To stop the program execution
C) To return a value to the operating system
D) To take input from the user

Answer: C) To return a value to the operating system

Explanation: In C, return 0; indicates successful execution of the main() function.

Q6. What is the correct syntax for a for loop in C?

A) for (int i = 1; i <= 5; i++)
B) for i in range(1,5):
C) loop (int i = 1; i <= 5; i++)
D) repeat for (int i = 1; i <= 5; i++)

Answer: A) for (int i = 1; i <= 5; i++)

Explanation: The correct C syntax for a loop follows: for (initialization; condition; update).

Q7. What will be the output of this C program?

```
#include <stdio.h>
int main() {
int x = 10;
x += 5;
printf("%d", x);
return 0;
}
```

A) 5
B) 10
C) 15
D) 20

Answer: C) 15

Explanation: x += 5; is equivalent to x = x + 5;, so 10 + 5 = 15.

Q8. What is XML mainly used for?

A) Styling web pages
B) Storing and transporting data
C) Creating animations
D) Writing C programs

Answer: B) Storing and transporting data

Explanation: XML is designed to store, structure, and transport data in a readable format.

Q9. What is the correct XML syntax for an element?

A) <element></element>
B) <element>
C) <element> </close>
D) element: </element>

Answer: A) <element></element>

Explanation: XML elements must have an opening tag and a closing tag.

Q10. Which of the following is NOT true about XML?

A) XML is case-sensitive
B) XML tags must be properly nested
C) XML can store data in a hierarchical format
D) XML is used for designing web page layouts

Answer: D) XML is used for designing web page layouts

Explanation: XML is used for data storage and transfer, not designing pages (HTML & CSS are used for that).

Q11. What is the root element in this XML document?

```
<students>
```

```
<student>
<name>Alice</name>
</student>
</students>
```

A) <student>
B) <name>
C) <students>
D) Alice

Answer: C) <students>

Explanation: The root element is the topmost element that contains all other elements.

Q12. Which of the following statements about XML attributes is correct?

A) XML attributes store extra information about an element
B) XML attributes must be written in capital letters
C) XML attributes can be used only in HTML
D) XML attributes are always numeric

Answer: A) XML attributes store extra information about an element

Explanation: XML attributes provide additional details about an element (e.g., <student age="12">).

Q13. What is the main purpose of Turtle in Python?

A) To write HTML code
B) To create simple graphics and drawings
C) To manage databases
D) To create mobile applications

Answer: B) To create simple graphics and drawings

Explanation: Python's Turtle module is used for drawing and graphics.

Q14. Which command moves the turtle forward by 100 units?

A) turtle.move(100)
B) t.forward(100)
C) t.jump(100)
D) t.go(100)

Answer: B) t.forward(100)

Explanation: The correct Turtle command to move forward is t.forward().

Q15. What will this Turtle Python code draw?

```
import turtle
t = turtle.Turtle()
t.forward(100)
t.left(90)
t.forward(100)
```

A) A straight line
B) A right angle
C) A circle
D) A rectangle

Answer: B) A right angle

Explanation: The code moves forward by 100, then turns left 90 degrees and moves forward again.

Section 2: Advanced Coding Concepts

Q16. What will be the output of the following C program?

```
#include <stdio.h>
int main() {
int a = 5, b = 2;
float result;
result = a / b;
printf("%f", result);
return 0;
}
```

A) 2.5
B) 2.000000
C) 2
D) Compilation error

Answer: B) 2.000000

Explanation: In C, a / b (where both a and b are integers) performs integer division, resulting in 2. Since result is float, it prints as 2.000000.

Q17. What does this C program output?

```
#include <stdio.h>
void function(int *a) {
*a = *a + 10;
}
int main() {
int x = 5;
function(&x);
printf("%d", x);
return 0;
}
```

A) 5
B) 10
C) 15
D) Compilation error

Answer: C) 15

Explanation: The function modifies x using call by reference, adding 10 to its value.

Q18. What is the output of this bitwise operation in C?

```
#include <stdio.h>
int main() {
int x = 5, y = 3;
printf("%d", x & y);
return 0;
}
```

A) 7
B) 1
C) 2
D) 5

Answer: B) 1

Explanation: The bitwise AND operation: 5 = 101 3 = 011 101 & 011 = 001 (1 in decimal)

Q19. Which of the following is true about XML and databases?

A) XML cannot be used to store data in databases.
B) XML data is always stored in a single file.
C) XML can be used as a format for exchanging structured data between databases.
D) XML requires special programming languages to be used.

Answer: C

Explanation: XML is commonly used to exchange structured data between systems, including database-driven applications, because it can represent hierarchical data in a platform-independent format.

Q20. What will happen if you try to parse this XML document?

```
<students>
<student id="1">
<name>Alice</name>
</student>
<student id="2">
<name>Bob</name>
</students>
```

A) It will parse successfully
B) It will cause an error due to incorrect nesting
C) It will ignore the error and display data correctly
D) It will create a new <student> tag automatically

Answer: B) It will cause an error due to incorrect nesting

Explanation: The closing </student> tag for Bob is missing, making the XML malformed.

Q21. What is the correct way to define an attribute in XML?

A) <book id=123>Title</book>
B) <book id='123'>Title</book>
C) <book>id="123"</book>
D) <book id>123</book>

Answer: B) <book id='123'>Title</book>

Explanation: XML attributes should be inside the opening tag and enclosed in quotes.

Q22. What will the following Turtle program draw?

```
import turtle
t = turtle.Turtle()
for i in range(4):
t.forward(100)
t.left(90)
t.circle(50)
```

A) A triangle inside a circle
B) A square followed by a circle
C) A hexagon inside a square
D) A single straight line

Answer: B) A square followed by a circle

Explanation: The loop draws a square, and t.circle(50) draws a circle next.

Q23. What will happen if you run this Turtle code?

```
import turtle
t = turtle.Turtle()
t.goto(50, 50)
```

```
t.pendown()
t.goto(-50, 50)
t.penup()
t.goto(50, -50)
t.pendown()
t.goto(-50, -50)
```

A) Two diagonal lines
B) Two horizontal lines
C) A rectangle
D) A single vertical line

Answer: B

Explanation: The first line goes horizontally from (50, 50) to (-50, 50). The pen is lifted before moving to (50, -50), then lowered to draw a second horizontal line to (-50, -50).

Q24. What will be the output of the following Turtle command?

```
t.color("blue", "yellow")
t.begin_fill()
t.circle(50)
t.end_fill()
```

A) A blue-filled circle
B) A yellow-filled circle with a blue outline
C) A completely yellow circle without an outline
D) No circle is drawn

Answer: B) A yellow-filled circle with a blue outline

Explanation: The first argument in t.color() sets the outline color, while the second argument sets the fill color.

Q25. What will this Turtle function do?

```
def draw_shape(sides, length):
for _ in range(sides):
t.forward(length)
t.left(360 / sides)
```

A) Draws a straight line
B) Draws a star
C) Draws a polygon with sides number of sides
D) Draws a circle

Answer: C) Draws a polygon with sides number of sides

Explanation: The loop moves length and turns by 360 / sides, forming a polygon.

Q26. Which C library is commonly used for handling graphics in game development?

A) stdio.h
B) stdlib.h
C) conio.h
D) graphics.h

Answer: D) graphics.h

Explanation: The graphics.h library provides functions for drawing shapes, handling colors, and managing pixels, making it useful for game development in C.

Q27. What is the purpose of using the rand() function in C for game development?

A) To generate random numbers for game elements like enemies or dice rolls
B) To restart the game when it crashes

C) To store game scores permanently
D) To control user input

Answer: **A) To generate random numbers for game elements like enemies or dice rolls**

Explanation: The rand() function in C is used to generate random numbers, which is essential in games for random enemy placement, power-ups, or dice rolls.

Q28. What will the following C program output?

```
#include <stdio.h>
int main() {
int x = 3, y = 5, temp;
temp = x;
x = y;
y = temp;
printf("%d %d", x, y);
return 0;
}
```

A) 3 5
B) 5 3
C) 3 3
D) 5 5

Answer: **B) 5 3**

Explanation: This program swaps x and y using a temporary variable, a common technique in game mechanics like character movement swaps.

Q29. How is XML used in game development?

A) To store high scores in databases
B) To define game levels, settings, and character properties
C) To execute Python game scripts
D) To replace the main game loop

Answer: **B) To define game levels, settings, and character properties**

Explanation: XML is used in game development to store structured data, such as level configurations, character attributes, and game settings.

Q30. What does this XML code represent in a game?

```
<character name="hero">
<health>100</health>
<attack>20</attack>
<speed>10</speed>
</character>
```

A) The code will crash the game
B) It represents an image file for a hero character
C) It defines the hero character's stats in a game
D) It is a Python script for controlling the character

Answer: **C) It defines the hero character's stats in a game**

Explanation: XML is commonly used to store character attributes like health, attack power, and speed, making game development easier and structured.

Q31. What is the advantage of using XML for game development over hardcoding values?

A) XML allows modifying game settings without changing the game code
B) XML slows down the game but improves security

C) XML is only used in large-scale AAA games
D) XML makes the game completely unhackable

Answer: A) XML allows modifying game settings without changing the game code

Explanation: XML separates data from logic, allowing developers to change game configurations without modifying the core game code.

Q32. What will the following Python Turtle code do?

```
import turtle
t = turtle.Turtle()
t.forward(100)
t.right(90)
t.forward(100)
```

A) Draws a straight line
B) Draws a right-angled path
C) Draws a square
D) Creates a game loop

Answer: B) Draws a right-angled path

Explanation: The turtle moves 100 steps forward, turns 90 degrees right, and moves 100 steps again, forming an L-shape.

Q33. Which of the following is the correct way to detect keypresses in a Python Turtle-based game?

A) turtle.keypress("space", jump)
B) screen.onkeypress(jump, "space")
C) screen.detectkeypress(jump, "space")
D) turtle.detect("space", jump)

Answer: B) screen.onkeypress(jump, "space")

Explanation: The onkeypress() function binds a key (like "space") to a function (like jump), allowing players to control their game character.

Q34. What does this Python Turtle game function do?

```
def move_up():
t.setheading(90)
t.forward(20)
```

A) Moves the turtle downwards
B) Moves the turtle upwards by 20 units
C) Rotates the turtle without movement
D) Moves the turtle in a random direction

Answer: B) Moves the turtle upwards by 20 units

Explanation: setheading(90) rotates the turtle to face upwards, and forward(20) moves it up by 20 units.

Q35. Which of the following is NOT a common feature of a game loop?

A) Rendering game graphics
B) Handling player input
C) Modifying game physics
D) Executing SQL queries

Answer: D) Executing SQL queries

Explanation: A game loop continuously updates the game state, but SQL queries are not typically part of real-time game rendering.

Q36. What will be the output of this C program?

```
#include <stdio.h>
int main() {
int a = 5;
printf("%d %d", a++, ++a);
return 0;
}
```

A) 5 6
B) 6 6
C) 5 7
D) Undefined behavior

Answer: D) Undefined behavior

Explanation: The expression printf("%d %d", a++, ++a); modifies a twice without a sequence point, leading to undefined behavior.

Q37. What is the correct syntax to declare a pointer in C?

A) int ptr;
B) int *ptr;
C) ptr *int;
D) pointer<int> ptr;

Answer: B

Explanation: The string literal "hello" requires six characters including the null terminator. A char array of size 5 is not large enough for a valid C string initialization.

Q38. What will the following program print?

```
#include <stdio.h>
int main() {
int arr[] = {10, 20, 30, 40};
printf("%d", *(arr + 2));
return 0;
}
```

A) 10
B) 20
C) 30
D) 40

Answer: C) 30

Explanation: arr + 2 moves two positions ahead, so *(arr + 2) accesses the third element (30).

Q39. What does sizeof() return when applied to an integer variable?

A) Size of the integer in bits
B) Size of the integer in bytes
C) Always 4
D) Always 8

Answer: B) Size of the integer in bytes

Explanation: sizeof(int) typically returns 4 bytes on most systems, but it depends on the architecture.

Q40. What is the purpose of fopen() in C?

A) Opens a file for reading and writing
B) Creates an array

C) Declares a function
D) Runs an infinite loop

Answer: A) Opens a file for reading and writing

Explanation: fopen(filename, mode) opens a file in the specified mode ("r", "w", "a", etc.).

Q41. Which of the following is NOT a feature of XML?

A) Human-readable
B) Self-descriptive
C) Fixed set of tags
D) Used to structure data

Answer: C) Fixed set of tags

Explanation: Unlike HTML, XML allows users to define their own tags.

Q42. What is the root element in this XML document?

```
<game>
<player>John</player>
<score>2500</score>
</game>
```

A) game
B) player
C) score
D) None of the above

Answer: A) game

Explanation: The root element is the outermost tag enclosing all elements.

Q43. How do you write an XML comment?

A) // This is a comment
B) <!-- This is a comment -->
C) /* This is a comment */
D) { This is a comment }

Answer: B) <!-- This is a comment -->

Explanation: XML uses <!-- --> for comments.

Q44. What is the purpose of an XML Schema (XSD)?

A) To define the structure and data types of XML documents
B) To store images
C) To write Python code
D) To add styles like CSS

Answer: A) To define the structure and data types of XML documents

Explanation: XML Schema (XSD) ensures XML data follows specific rules and formats.

Q45. What will the following Python Turtle code do?

```
import turtle
t = turtle.Turtle()
t.circle(50)
```

A) Draw a circle with radius 50
B) Move forward by 50 units
C) Draw a square
D) Show an error

Answer: **A) Draw a circle with radius 50**

Explanation: circle(50) draws a circle of radius 50 pixels.

Q46. What will happen if you run the following Turtle program?

```
import turtle
t = turtle.Turtle()
t.forward(100)
t.right(90)
t.forward(100)
```

A) The turtle moves 100 steps forward and turns right

B) The turtle moves in a circle

C) The turtle moves randomly

D) The program crashes

Answer: **A) The turtle moves 100 steps forward and turns right**

Explanation: forward(100) moves the turtle 100 pixels, and right(90) turns it right by 90 degrees.

Q47. What does this Turtle code do?

```
import turtle
t = turtle.Turtle()
for i in range(4):
t.forward(100)
t.right(90)
```

A) Draws a circle

B) Draws a square

C) Moves in a zigzag pattern

D) Moves forward without turning

Answer: **B) Draws a square**

Explanation: The loop moves 100 steps and turns 90° four times, forming a square.

Q48. What is the output of this Turtle code?

```
import turtle
t = turtle.Turtle()
t.color("blue")
t.forward(50)
```

A) Turtle moves forward and changes to blue

B) Turtle moves forward and changes to red

C) Turtle moves backward

D) Turtle moves forward but remains black

Answer: **A) Turtle moves forward and changes to blue**

Explanation: t.color("blue") changes the pen color to blue.

Q49. What does t.speed(0) do in Turtle programming?

A) Stops the turtle

B) Makes the turtle move fastest

C) Slows down the turtle

D) Moves the turtle in random directions

Answer: **B) Makes the turtle move fastest**

Explanation: t.speed(0) sets the fastest drawing speed in Turtle.

Q50. Case Study: Game Development using C Programming and XML

You are tasked with developing a simple text-based game where a player can input their name and score, and the game will display a summary of their performance in a leaderboard. To do this, you'll need to:

Write a C program that accepts the player's name and score.

Store the data in a structured format, using XML to save the leaderboard.

The game must also use Turtle programming to display a graphical leaderboard where the scores are visualized in a basic bar graph format.

What does the following C code do?

```
#include <stdio.h>
#include <string.h>
int main() {
char name[50];
int score;
printf("Enter player name: ");
scanf("%s", name);
printf("Enter score: ");
scanf("%d", &score);
printf("\nPlayer: %s\nScore: %d\n", name, score);
return 0;
}
```

A) It stores player data in an XML file.

B) It accepts a player's name and score, then displays them.

C) It calculates the player's score.

D) It draws a graphical representation of the player's score.

Answer: B) It accepts a player's name and score, then displays them.

Explanation: This C code allows the user to input the player's name and score, then prints the name and score.

School Connect Online

Compact Answer Key

Q.No	Answer	Section
1	B	Section 1: Programming Basics
2	B	Section 1: Programming Basics
3	A	Section 1: Programming Basics
4	B	Section 1: Programming Basics
5	C	Section 1: Programming Basics
6	A	Section 1: Programming Basics
7	C	Section 1: Programming Basics
8	B	Section 1: Programming Basics
9	A	Section 1: Programming Basics
10	D	Section 1: Programming Basics
11	C	Section 1: Programming Basics
12	A	Section 1: Programming Basics
13	B	Section 1: Programming Basics
14	B	Section 1: Programming Basics
15	B	Section 1: Programming Basics
16	B	Section 2: Advanced Coding Concepts
17	C	Section 2: Advanced Coding Concepts
18	B	Section 2: Advanced Coding Concepts
19	C	Section 2: Advanced Coding Concepts
20	B	Section 2: Advanced Coding Concepts
21	B	Section 2: Advanced Coding Concepts
22	B	Section 2: Advanced Coding Concepts
23	B	Section 2: Advanced Coding Concepts
24	B	Section 2: Advanced Coding Concepts
25	C	Section 2: Advanced Coding Concepts
26	D	Section 2: Advanced Coding Concepts
27	A	Section 2: Advanced Coding Concepts
28	B	Section 2: Advanced Coding Concepts
29	B	Section 2: Advanced Coding Concepts
30	C	Section 2: Advanced Coding Concepts
31	A	Section 2: Advanced Coding Concepts
32	B	Section 2: Advanced Coding Concepts
33	B	Section 2: Advanced Coding Concepts
34	B	Section 2: Advanced Coding Concepts
35	D	Section 2: Advanced Coding Concepts
36	D	Section 2: Advanced Coding Concepts
37	B	Section 2: Advanced Coding Concepts
38	C	Section 2: Advanced Coding Concepts
39	B	Section 2: Advanced Coding Concepts
40	A	Section 2: Advanced Coding Concepts
41	C	Section 2: Advanced Coding Concepts
42	A	Section 2: Advanced Coding Concepts
43	B	Section 2: Advanced Coding Concepts
44	A	Section 2: Advanced Coding Concepts
45	A	Section 2: Advanced Coding Concepts
46	A	Section 2: Advanced Coding Concepts
47	B	Section 2: Advanced Coding Concepts
48	A	Section 2: Advanced Coding Concepts
49	B	Section 2: Advanced Coding Concepts
50	B	Section 2: Advanced Coding Concepts