Grade 13 A/L ICT

Lesson 9: Programming

- 9.1 The Problem-Solving Process
- 9.2 Top-Down Design, Stepwise Refinement and Structure Charts
- 9.3 Algorithms: Flowcharts, Pseudocode and Hand Traces
- 9.4 Programming Languages and Paradigms
- 9.5 Program Translation: Compilers, Interpreters and Linkers
- 9.6 Integrated Development Environments and Debugging
- 9.7 Python Basics: Structure, Variables, Data Types, Operators and I/O
- 9.8 Control Structures: Sequence, Selection and Repetition
- 9.9 Functions: Built-in and User-defined
- 9.10 Data Structures: Strings, Lists, Tuples and Dictionaries
- 9.11 File Handling: Open, Read, Write, Append and Close
- 9.12 Python and Databases: Connect, Add, Modify, Delete, Retrieve
- 9.13 Searching and Sorting: Sequential Search and Bubble Sort

© 2026 schoolict.net (Ruwan Mangala Suraweera)
All rights reserved.

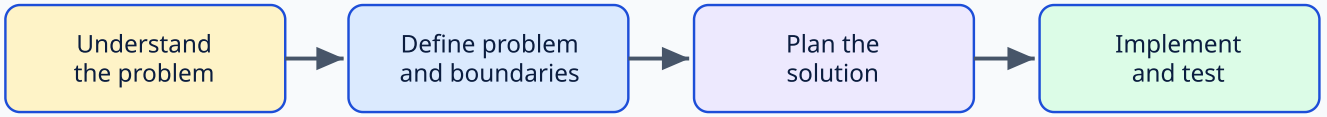
Grade 13 A/L ICT · Lesson 9

9.1 The Problem-Solving Process

Syllabus 9.1. Before writing code, a programmer solves the problem on paper. The four steps are: **understand the problem**, **define the problem and its boundaries**, **plan the solution** and **implement** it.

1. The four steps

Problem-solving process



Input → process → output: list what is given, what is needed and the limits first.

Problem-solving steps

Understand
Read carefully: what is given (input) and what is wanted (output)?

Define and set boundaries
State the exact task and limits, e.g. lengths are positive numbers in metres.

Plan
Choose the steps (algorithm) as a flowchart or pseudocode.

Implement
Write the program, run it with test data, fix errors and evaluate whether it really solves the problem.

Part	Rectangle problem
Input	length, width
Process	area = length × width; perimeter = 2 × (length + width)
Output	area, perimeter
Boundary	length and width > 0

2. Implementing the plan in Python

🐍 Rectangle area and perimeter

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

```
# 1 input  2 process  3 output
length = float(input("Length: "))
width = float(input("Width: "))
area = length * width
perimeter = 2 * (length + width)
print("Area =", area)
print("Perimeter =", perimeter)
```

▶ Output

```
Length: 5
Width: 3
Area = 15.0
Perimeter = 16.0
```

🧱 **Worked example: Test the rectangle program with length 5 and width 3.**
Hand calculation: area = 5 × 3 = **15**, perimeter = 2 × (5 + 3) = **16**. The program printed Area = 15.0 and Perimeter = 16.0, so the plan is correct.

💡 Always work out the expected answer by hand before you run the program.

🚀 **Beyond the syllabus (2026 update)** *Not required for the exam – read for interest.* In 2026 AI assistants can write code from a problem statement, but a clear definition and test cases are still the programmer's job.

Key Words

Problem solving – Process of finding a method to reach a goal.

Understanding the problem – Finding what is given and what is required.

Problem definition – Exact statement of the task to be solved.

Boundary – Limit on the scope or allowed values of a problem.

Planning – Designing the steps of the solution before coding.

Implementation – Coding, running and testing the planned solution.

Input – Data given to the solution.

Process – Steps that change input into output.

Output – Result produced by the solution.

Test data – Input values used to check a program.

Expected result – Answer worked out by hand before running.

Algorithm – Finite ordered steps that solve a problem.

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

Sample Q&A

1. List the four steps of problem solving.

> **Answer:** Understand, define with boundaries, plan, implement.

2. What is a boundary of a problem?

> **Answer:** A limit such as “length must be greater than 0”.

3. Inputs of the rectangle problem?

> **Answer:** length and width.

4. Output for 5 and 3?

> **Answer:** Area 15.0, perimeter 16.0.

5. Why plan before coding?

> **Answer:** Fewer errors and a clear, checkable set of steps.

6. What happens in implementation?

> **Answer:** Code is written, run with test data and debugged.

MCQ

1. The first step of problem solving is

A. printing output
B. writing code
C. understanding the problem
D. choosing a language
E. testing

2. “Length must be positive” is a

A. process
B. test result
C. output
D. comment
E. boundary of the problem

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

3. Choosing the steps as a flowchart happens in

A. planning
B. understanding
C. maintenance
D. output
E. implementation

4. For length 5 and width 3 the area printed is

A. 16.0
B. 8.0
C. 2.0
D. 15.0
E. 30.0

5. For length 5 and width 3 the perimeter printed is

A. 30.0
B. 8.0
C. 16.0
D. 15.0
E. 10.0

6. Running the program with sample values is part of

A. boundary setting
B. implementation
C. planning only
D. understanding
E. defining

7. In the rectangle problem, length and width are the

A. inputs
B. outputs
C. comments
D. constants only
E. boundaries

8. An answer worked out by hand before running is the

A. boundary
B. actual output
C. expected result
D. syntax error
E. variable

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

9. Which term means: “Process of finding a method to reach a goal”?

A. Problem definition
B. Expected result
C. Process
D. Implementation
E. Problem solving

10. Which term means: “Finding what is given and what is required”?

A. Boundary
B. Understanding the problem
C. Input
D. Algorithm
E. Output

11. Which term means: “Exact statement of the task to be solved”?

A. Test data
B. Process
C. Problem solving
D. Planning
E. Problem definition

12. Which term means: “Limit on the scope or allowed values of a problem”?

A. Output
B. Understanding the problem
C. Expected result
D. Boundary
E. Implementation

13. Which term means: “Designing the steps of the solution before coding”?

A. Test data
B. Input
C. Algorithm
D. Planning
E. Problem definition

14. Which term means: “Coding, running and testing the planned solution”?

A. Implementation
B. Process
C. Expected result
D. Problem solving
E. Boundary

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

15. Which term means: “Data given to the solution”?

A. Input
B. Algorithm
C. Planning
D. Understanding the problem
E. Output

16. Which term means: “Steps that change input into output”?

A. Test data
B. Problem definition
C. Implementation
D. Problem solving
E. Process

17. Which term means: “Result produced by the solution”?

A. Input
B. Output
C. Expected result
D. Understanding the problem
E. Boundary

18. Which term means: “Input values used to check a program”?

A. Problem definition
B. Planning
C. Test data
D. Algorithm
E. Process

19. Which term means: “Answer worked out by hand before running”?

A. Problem solving
B. Expected result
C. Output
D. Implementation
E. Boundary

20. Which term means: “Finite ordered steps that solve a problem”?

A. Test data
B. Input
C. Understanding the problem
D. Algorithm
E. Planning

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

Structured Questions

1. Describe the four steps of problem solving. *(4 marks)*

> Understand; define and boundaries; plan; implement and test.

2. Identify input, process and output for the rectangle problem. *(3 marks)*

> Input length, width; area = l × w and perimeter = 2(l + w); output both.

3. State two boundaries for a marks-entry problem. *(2 marks)*

> Marks 0–100; whole numbers only.

4. Why should expected results be found before running? *(2 marks)*

> Compare actual output; detect logic errors.

5. Write a Python program to find area and perimeter of a rectangle. *(3 marks)*

> input with float; area and perimeter formulas; print both.

Summary

- Understand the problem first.
- Define the task and its boundaries.
- Plan the steps before coding.
- Implement and test with data.
- Compare output with the expected result.

MCQ answer key – 9.1

1. **C** – Understand first.
2. **E** – Limit.
3. **A** – Plan.
4. **D** – 5 × 3.
5. **C** – 2 × 8.
6. **B** – Implement and test.
7. **A** – Inputs.
8. **C** – Expected.
9. **E** – Problem solving – Process of finding a method to reach a goal.
10. **B** – Understanding the problem – Finding what is given and what is required.
11. **E** – Problem definition – Exact statement of the task to be solved.
12. **D** – Boundary – Limit on the scope or allowed values of a problem.
13. **D** – Planning – Designing the steps of the solution before coding.
14. **A** – Implementation – Coding, running and testing the planned solution.
15. **A** – Input – Data given to the solution.

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

16. **E** – Process – Steps that change input into output.
17. **B** – Output – Result produced by the solution.
18. **C** – Test data – Input values used to check a program.
19. **B** – Expected result – Answer worked out by hand before running.
20. **D** – Algorithm – Finite ordered steps that solve a problem.

© 2026 schoolict.net (Ruwan Mangala Suraweera). All rights reserved.

Grade 13 A/L ICT · Lesson 9

9.2 Top-Down Design, Stepwise Refinement and Structure Charts

Syllabus 9.2. Big problems are easier when split into smaller parts. **Modularisation** divides a program into modules; **top-down design** starts with the whole task and **stepwise refinement** breaks each step into smaller steps until each can be coded. A **structure chart** shows the modules and how they call each other.

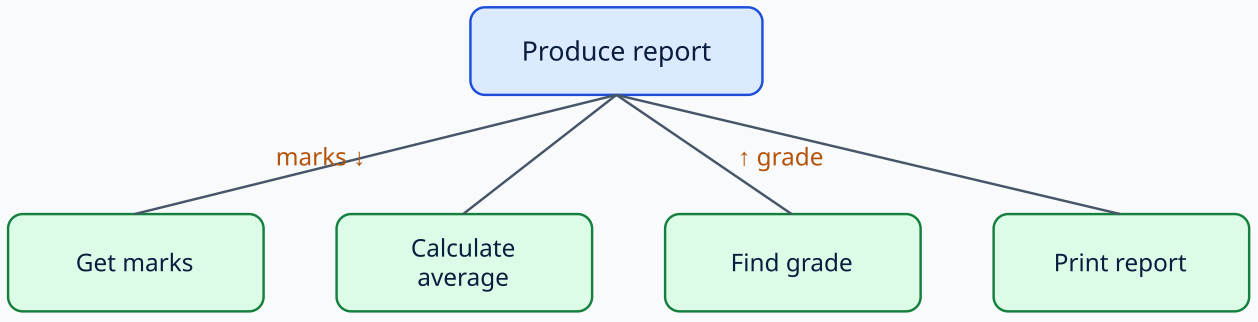
1. Stepwise refinement

Refining the task step by step

```
1 Produce a class report
  1.1 Get the marks
  1.2 Calculate the average
      1.2.1 Add all marks
      1.2.2 Divide by the number of marks
  1.3 Find the grade
      1.3.1 IF average >= 75 THEN A
      1.3.2 ELSE IF average >= 50 THEN C ELSE F
  1.4 Print the report
```

2. Structure chart

Structure chart (top-down design)



Each box is a module; lines show which module calls which; arrows on lines show data passed.

Structure chart for the class report

- Each **rectangle** is a module (a function in Python).
- **Lines** join a calling module to the modules it calls.
- Small arrows show **data** passed down (marks) and back up (grade).

3. Each module becomes a function

🐍 Top-down: a report built from small functions

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

```
def get_marks():
    return [78, 45, 88, 62, 39]

def average(marks):
    return sum(marks) / len(marks)

def grade(avg):
    if avg >= 75:
        return "A"
    elif avg >= 50:
        return "C"
    return "F"

def report():
    marks = get_marks()
    avg = average(marks)
    print("Average:", avg)
    print("Grade:", grade(avg))

report()
```

▶ Output

```
Average: 62.4
Grade: C
```

🧱 **Worked example: What does the modular program print, and which module decides it?** average() returns (78 + 45 + 88 + 62 + 39) ÷ 5 = **62.4**. grade() sees 62.4 ≥ 50 but < 75, so it returns **C**. report() prints both.

💡 Benefits of modules: easier testing, reuse, teamwork and simpler maintenance.

🚀 **Beyond the syllabus (2026 update)** *Not required for the exam – read for interest.* Large 2026 apps are split into microservices – the same top-down idea at the scale of whole servers.

Key Words

Modularisation – Dividing a program into separate modules.

Module – Self-contained part of a program that does one task.

Top-down design – Starting with the whole problem and dividing it into parts.

Stepwise refinement – Breaking each step into smaller, more detailed steps.

Structure chart – Diagram of modules and their calling hierarchy.

Main module – Top module that controls the sub-modules.

Sub-module – Module called by a higher module.

Data couple – Arrow on a structure chart showing data passed.

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

Reuse – Using the same module in more than one place.

Function – Named block of code that can be called.

Bottom-up design – Building small parts first and combining them.

Decomposition – Breaking a complex problem into simpler ones.

Sample Q&A

1. What is stepwise refinement?

> **Answer:** Breaking each step into smaller steps until each can be coded.

2. What does a structure chart show?

> **Answer:** Modules, who calls whom and data passed.

3. Two benefits of modularisation?

> **Answer:** Easier testing and reuse.

4. Average in the modular program?

> **Answer:** 62.4.

5. Grade printed?

> **Answer:** C.

6. Top-down vs bottom-up?

> **Answer:** Top-down splits the whole; bottom-up builds parts then combines.

MCQ

1. Starting from the whole problem and splitting it is

A. top-down design
B. compiling
C. hand tracing
D. bottom-up design
E. debugging

2. Breaking steps into smaller detailed steps is

A. sorting
B. stepwise refinement
C. normalisation
D. interpretation
E. linking

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

3. A diagram of modules and calls is a

A. structure chart
B. trace table
C. flowchart symbol
D. Gantt chart
E. ER diagram

4. In Python a module of a structure chart is usually written as a

A. tuple
B. comment
C. function
D. constant
E. file mode

5. The modular program prints average

A. 60.4
B. 62.4
C. 65.0
D. 62.0
E. 78.0

6. The modular program prints grade

A. S
B. F
C. A
D. C
E. B

7. Arrows beside the lines of a structure chart show

A. data passed between modules
B. loop counters
C. file names
D. comments
E. syntax errors

8. Which is NOT a benefit of modularisation?

A. easier maintenance
B. reuse of code
C. teamwork
D. easier testing
E. it always makes programs run faster

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

9. Which term means: “Dividing a program into separate modules”?

A. Modularisation
B. Data couple
C. Main module
D. Top-down design
E. Bottom-up design

10. Which term means: “Self-contained part of a program that does one task”?

A. Decomposition
B. Module
C. Stepwise refinement
D. Reuse
E. Sub-module

11. Which term means: “Starting with the whole problem and dividing it into parts”?

A. Modularisation
B. Function
C. Structure chart
D. Top-down design
E. Data couple

12. Which term means: “Breaking each step into smaller, more detailed steps”?

A. Main module
B. Bottom-up design
C. Module
D. Reuse
E. Stepwise refinement

13. Which term means: “Diagram of modules and their calling hierarchy”?

A. Sub-module
B. Top-down design
C. Decomposition
D. Function
E. Structure chart

14. Which term means: “Top module that controls the sub-modules”?

A. Stepwise refinement
B. Main module
C. Data couple
D. Bottom-up design
E. Modularisation

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

15. Which term means: “Module called by a higher module”?

A. Decomposition
B. Structure chart
C. Reuse
D. Sub-module
E. Module

16. Which term means: “Arrow on a structure chart showing data passed”?

A. Modularisation
B. Top-down design
C. Data couple
D. Main module
E. Function

17. Which term means: “Using the same module in more than one place”?

A. Sub-module
B. Bottom-up design
C. Stepwise refinement
D. Reuse
E. Module

18. Which term means: “Named block of code that can be called”?

A. Decomposition
B. Top-down design
C. Function
D. Data couple
E. Structure chart

19. Which term means: “Building small parts first and combining them”?

A. Modularisation
B. Stepwise refinement
C. Bottom-up design
D. Reuse
E. Main module

20. Which term means: “Breaking a complex problem into simpler ones”?

A. Function
B. Structure chart
C. Sub-module
D. Module
E. Decomposition

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

Structured Questions

1. Explain top-down design and stepwise refinement. *(3 marks)*

> Whole to parts; refine each step; until codable.

2. Refine "calculate average" into smaller steps. *(3 marks)*

> Add all marks; count them; divide total by count.

3. Draw a structure chart for producing a class report. *(3 marks)*

> Main Produce report; Get marks, Calculate average, Find grade, Print report; data couples.

4. State three benefits of modularisation. *(3 marks)*

> Testing; reuse; teamwork/maintenance.

5. Write a Python function grade(avg) returning A, C or F. *(4 marks)*

> def grade(avg); if avg >= 75 return A; elif avg >= 50 return C; return F.

Summary

- Modularisation splits programs into modules.
- Top-down: whole problem first.
- Stepwise refinement adds detail step by step.
- Structure charts show modules and calls.
- Each module becomes a function.

MCQ answer key – 9.2

1. **A** – Top-down.
2. **B** – Refinement.
3. **A** – Structure chart.
4. **C** – Function.
5. **B** – 312 ÷ 5.
6. **D** – 50 ≤ 62.4 < 75.
7. **A** – Data couples.
8. **E** – Speed is not guaranteed.
9. **A** – Modularisation – Dividing a program into separate modules.
10. **B** – Module – Self-contained part of a program that does one task.
11. **D** – Top-down design – Starting with the whole problem and dividing it into parts.
12. **E** – Stepwise refinement – Breaking each step into smaller, more detailed steps.
13. **E** – Structure chart – Diagram of modules and their calling hierarchy.
14. **B** – Main module – Top module that controls the sub-modules.
15. **D** – Sub-module – Module called by a higher module.

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

16. **C** – Data couple – Arrow on a structure chart showing data passed.
17. **D** – Reuse – Using the same module in more than one place.
18. **C** – Function – Named block of code that can be called.
19. **C** – Bottom-up design – Building small parts first and combining them.
20. **E** – Decomposition – Breaking a complex problem into simpler ones.

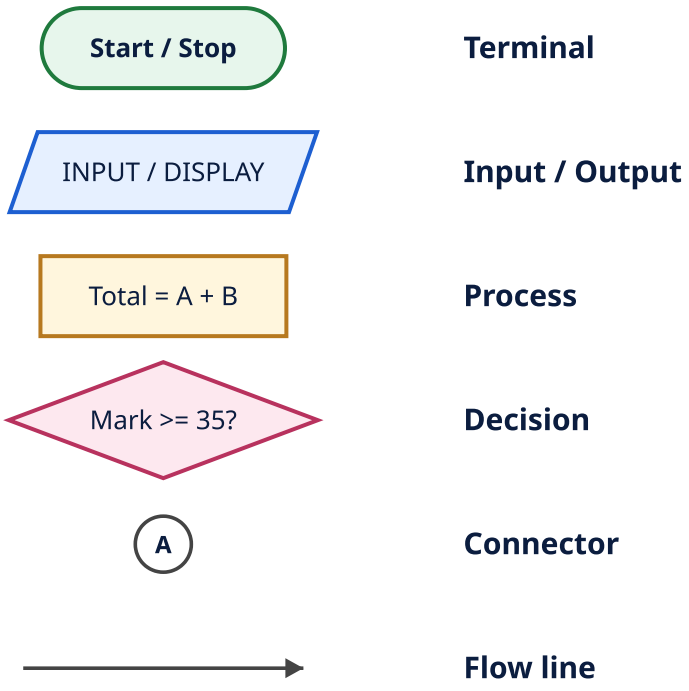
© 2026 schoolict.net (Ruwan Mangala Suraweera). All rights reserved.

Grade 13 A/L ICT · Lesson 9

9.3 Algorithms: Flowcharts, Pseudocode and Hand Traces

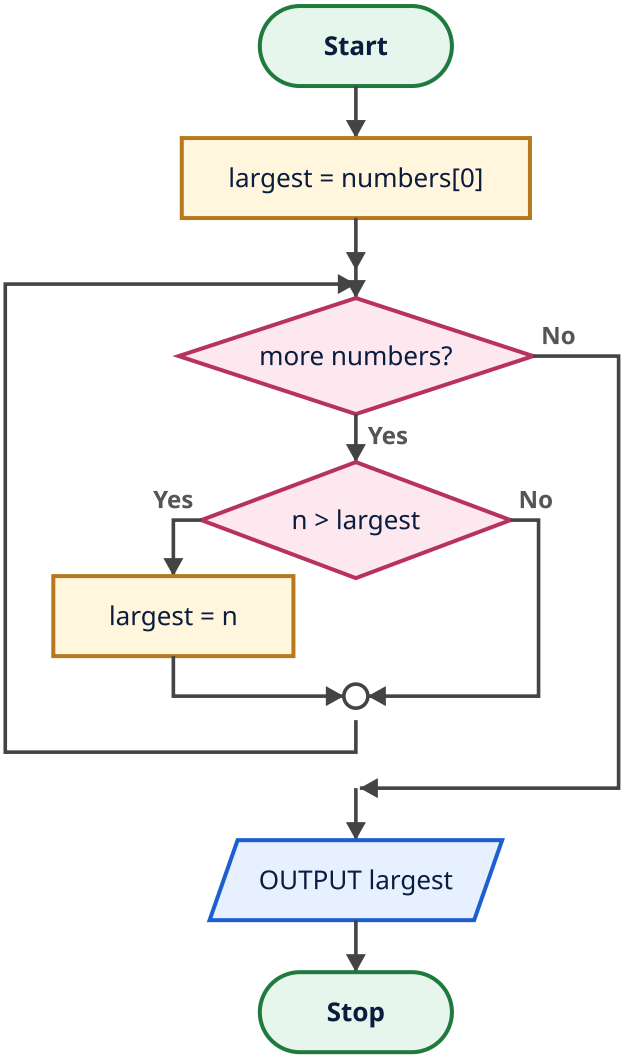
Syllabus 9.3. An **algorithm** is a finite, ordered set of clear steps that solves a problem. We show algorithms as **flowcharts** (standard symbols) or **pseudocode** (structured English), and check them with a **hand trace** (trace table) before coding.

1. Flowchart symbols



Standard flowchart symbols

2. Finding the largest number



Flowchart: find the largest number

Pseudocode

```
BEGIN
  largest = first number
  FOR each number n DO
    IF n > largest THEN
      largest = n
    ENDIF
  ENDFOR
  DISPLAY largest
END
```

3. Hand trace

n	n > largest?	largest
12	False	12
45	True	45

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

7	False	45
30	False	45

🧱 **Worked example: Trace the algorithm for numbers [12, 45, 7, 30].**
largest starts at 12. Only 45 is greater, so largest becomes 45; 7 and 30 are not. Output **45** – the Python program below prints the same.

🐍 Find the largest number

```
numbers = [12, 45, 7, 30]
largest = numbers[0]
for n in numbers:
    if n > largest:
        largest = n
print("Largest:", largest)
```

▶ Output

```
Largest: 45
```

💡 A trace table has one column per variable and one row each time a value changes or a condition is tested.

🚀 **Beyond the syllabus (2026 update)** *Not required for the exam – read for interest.* In 2026 visual debuggers draw live trace tables automatically, but exams still ask you to trace by hand.

Key Words

Algorithm – Finite ordered unambiguous steps to solve a problem.

Flowchart – Diagram of an algorithm using standard symbols.

Pseudocode – Structured English description of an algorithm.

Hand trace – Working through an algorithm by hand to check values.

Trace table – Table recording variable values during a trace.

Terminal symbol – Rounded box marking start or stop.

Process symbol – Rectangle for a calculation or assignment.

Decision symbol – Diamond with a yes/no condition.

Input/output symbol – Parallelogram for reading or displaying data.

Flow line – Arrow showing the order of steps.

Connector – Small circle joining parts of a flowchart.

Dry run – Another name for a hand trace without a computer.

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

Sample Q&A

1. Define an algorithm.

> **Answer:** Finite ordered clear steps that solve a problem.

2. Which symbol shows a decision?

> **Answer:** Diamond.

3. Which symbol shows input or output?

> **Answer:** Parallelogram.

4. Output for [12, 45, 7, 30]?

> **Answer:** 45.

5. How many times does largest change after the start?

> **Answer:** Once (12 → 45).

6. Why hand trace?

> **Answer:** To find logic errors before coding.

MCQ

1. A diamond in a flowchart shows a

A. connector
B. decision
C. terminal
D. input
E. process

2. A parallelogram shows

A. a connector
B. start or stop
C. input or output
D. a calculation only
E. a decision

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

3. A rounded rectangle shows

A. a loop counter
B. input
C. a decision
D. start or stop
E. a file

4. Structured English for an algorithm is

A. a trace table
B. pseudocode
C. bytecode
D. a flowchart
E. machine code

5. The largest of [12, 45, 7, 30] is

A. 94
B. 7
C. 30
D. 12
E. 45

6. In the trace, n > largest is True for

A. 12 only
B. all values
C. no value
D. 7 and 30
E. 45 only

7. A table of variable values while checking an algorithm is a

A. trace table
B. hash table
C. symbol table
D. data dictionary
E. truth table

8. An algorithm must be

A. random
B. unordered
C. finite and unambiguous
D. written only in Python
E. infinite

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

9. Which term means: “Finite ordered unambiguous steps to solve a problem”?

A. Algorithm
B. Decision symbol
C. Terminal symbol
D. Connector
E. Pseudocode

10. Which term means: “Diagram of an algorithm using standard symbols”?

A. Dry run
B. Flowchart
C. Process symbol
D. Hand trace
E. Input/output symbol

11. Which term means: “Structured English description of an algorithm”?

A. Pseudocode
B. Trace table
C. Algorithm
D. Decision symbol
E. Flow line

12. Which term means: “Working through an algorithm by hand to check values”?

A. Terminal symbol
B. Connector
C. Input/output symbol
D. Flowchart
E. Hand trace

13. Which term means: “Table recording variable values during a trace”?

A. Pseudocode
B. Process symbol
C. Flow line
D. Trace table
E. Dry run

14. Which term means: “Rounded box marking start or stop”?

A. Connector
B. Decision symbol
C. Hand trace
D. Terminal symbol
E. Algorithm

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

15. Which term means: “Rectangle for a calculation or assignment”?

A. Flowchart
B. Trace table
C. Process symbol
D. Dry run
E. Input/output symbol

16. Which term means: “Diamond with a yes/no condition”?

A. Decision symbol
B. Flow line
C. Terminal symbol
D. Pseudocode
E. Algorithm

17. Which term means: “Parallelogram for reading or displaying data”?

A. Hand trace
B. Flowchart
C. Input/output symbol
D. Process symbol
E. Connector

18. Which term means: “Arrow showing the order of steps”?

A. Trace table
B. Decision symbol
C. Pseudocode
D. Flow line
E. Dry run

19. Which term means: “Small circle joining parts of a flowchart”?

A. Terminal symbol
B. Connector
C. Algorithm
D. Input/output symbol
E. Hand trace

20. Which term means: “Another name for a hand trace without a computer”?

A. Flow line
B. Trace table
C. Flowchart
D. Process symbol
E. Dry run

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

Structured Questions

1. Draw and name five flowchart symbols. *(5 marks)*

> Terminal; process; decision; input/output; flow line/connector.

2. Write pseudocode to find the largest of a list. *(4 marks)*

> largest = first; FOR each n; IF n > largest THEN largest = n; DISPLAY.

3. Draw a flowchart to find the largest of a list. *(6 marks)*

> Start; set largest; loop decision; compare decision; output; stop.

4. Complete a trace table for [12, 45, 7, 30]. *(5 marks)*

> 12 F 12; 45 T 45; 7 F 45; 30 F 45; output 45.

5. State three properties of a good algorithm. *(3 marks)*

> Finite; unambiguous; ordered steps with input/output.

Summary

- An algorithm is a finite set of clear steps.
- Flowcharts use standard symbols.
- Pseudocode is structured English.
- Trace tables check values by hand.
- Trace before you code.

MCQ answer key – 9.3

1. **B** – Decision.
2. **C** – I/O.
3. **D** – Terminal.
4. **B** – Pseudocode.
5. **E** – 45.
6. **E** – Only 45.
7. **A** – Trace table.
8. **C** – Finite and clear.
9. **A** – Algorithm – Finite ordered unambiguous steps to solve a problem.
10. **B** – Flowchart – Diagram of an algorithm using standard symbols.
11. **A** – Pseudocode – Structured English description of an algorithm.
12. **E** – Hand trace – Working through an algorithm by hand to check values.
13. **D** – Trace table – Table recording variable values during a trace.
14. **D** – Terminal symbol – Rounded box marking start or stop.
15. **C** – Process symbol – Rectangle for a calculation or assignment.

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

16. **A** – Decision symbol – Diamond with a yes/no condition.
17. **C** – Input/output symbol – Parallelogram for reading or displaying data.
18. **D** – Flow line – Arrow showing the order of steps.
19. **B** – Connector – Small circle joining parts of a flowchart.
20. **E** – Dry run – Another name for a hand trace without a computer.

© 2026 schoolict.net (Ruwan Mangala Suraweera). All rights reserved.

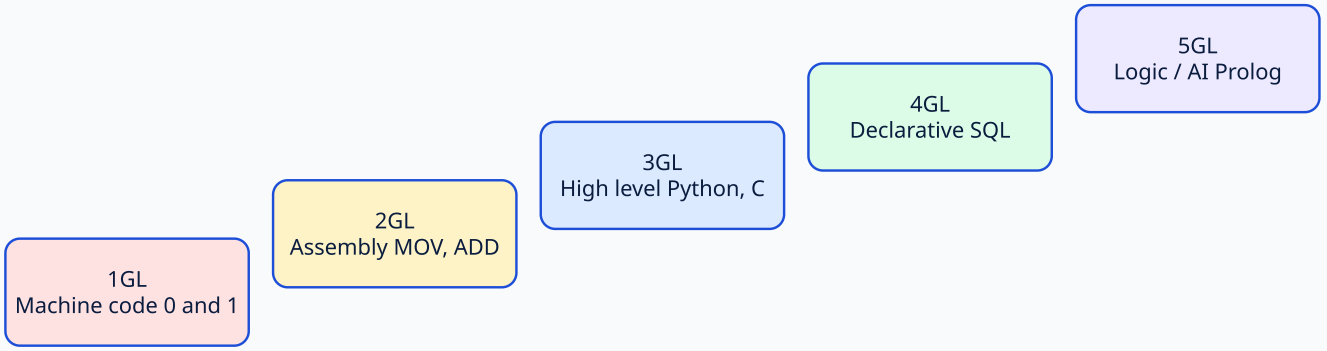
Grade 13 A/L ICT · Lesson 9

9.4 Programming Languages and Paradigms

Syllabus 9.4. Programming languages have evolved through **generations**, from machine code to languages close to human thinking. A **paradigm** is a style of programming: **imperative** (how, step by step), **declarative** (what result is wanted) and **object oriented** (objects that combine data and methods).

1. Generations

Generations of programming languages



Higher generations are closer to human language and further from the hardware.

Language generations 1GL–5GL

Generation	Example	Feature
1GL machine	10110000 01100001	Binary; runs directly; hard to write
2GL assembly	MOV AL, 61h	Mnemonics; needs an assembler
3GL high level	Python, C, Java	English-like; portable; compiled or interpreted
4GL	SQL	Say what you want; fewer lines
5GL	Prolog	Rules and logic for AI

2. Paradigms

Imperative
Commands change the program state step by step (C, Pascal, Python loops).

Declarative
Describe the result; the system decides how (SQL, Prolog, HTML).

Object oriented
Classes and objects with encapsulation, inheritance and polymorphism (Java, Python, C++).

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

🐍 One task in three styles

```
prices = [120, 80, 200]

# imperative: say HOW, step by step
total = 0
for p in prices:
    total = total + p
print("Imperative:", total)

# declarative style: say WHAT you want
print("Declarative:", sum(prices))

# object oriented: data + methods in a class
class Cart:
    def __init__(self, items):
        self.items = items
    def total(self):
        return sum(self.items)

print("OOP:", Cart(prices).total())
```

▶ Output

```
Imperative: 400
Declarative: 400
OOP: 400
```

🧱 **Worked example: The cart prices are 120, 80 and 200. Do the three styles agree?**
Hand total = 120 + 80 + 200 = **400**. The imperative loop, the declarative sum() and the Cart object all printed **400**.

💡 Python is multi-paradigm: it supports imperative, object oriented and some declarative styles.

🚀 **Beyond the syllabus (2026 update)** *Not required for the exam – read for interest.* In 2026 many developers describe what they want in plain language to an AI – an even more declarative style – but they still read and test the code.

Key Words

Machine language – First generation binary code run directly by the CPU.

Assembly language – Second generation language using mnemonics.

High-level language – Third generation English-like portable language.

Fourth generation language – Language where you state the result, such as SQL.

Fifth generation language – Logic and rule based language for AI such as Prolog.

Programming paradigm – A style or approach to writing programs.

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

Imperative paradigm – Style where statements change state step by step.	**Declarative paradigm** – Style that states what result is needed, not how.
Object oriented paradigm – Style built on objects combining data and methods.	**Class** – Template that defines attributes and methods of objects.
Object – Instance of a class holding its own data.	**Encapsulation** – Keeping data and methods together and hiding details.

Sample Q&A

1. List the five generations.

> **Answer:** Machine, assembly, high level, 4GL, 5GL.

2. Give a 4GL example.

> **Answer:** SQL.

3. Imperative vs declarative?

> **Answer:** Imperative says how; declarative says what.

4. Three OOP features?

> **Answer:** Encapsulation, inheritance, polymorphism.

5. What do all three styles print for the cart?

> **Answer:** 400.

6. Why is Python multi-paradigm?

> **Answer:** It supports imperative, OOP and declarative features.

MCQ

1. Binary code run directly by the CPU is

A. SQL
B. Prolog
C. machine language
D. assembly language
E. Python

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

2. A language using mnemonics such as MOV is

A. assembly language
B. Prolog
C. SQL
D. machine language
E. HTML

3. SQL belongs to the

A. fifth generation
B. first generation
C. second generation
D. third generation
E. fourth generation

4. Prolog is mainly

A. a 3GL
B. a logic / 5GL language
C. assembly
D. a markup language
E. machine code

5. Stating WHAT result is wanted rather than HOW is

A. assembly
B. machine level
C. declarative
D. imperative
E. procedural

6. A template that defines objects is a

A. compiler
B. loop
C. constant
D. class
E. tuple

7. In the paradigms program every style prints

A. 480
B. 120
C. 200
D. 400
E. 380

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

8. Hiding data inside an object with its methods is

A. normalisation
B. encapsulation
C. compilation
D. iteration
E. recursion

9. Which term means: “First generation binary code run directly by the CPU”?

A. High-level language
B. Declarative paradigm
C. Object
D. Programming paradigm
E. Machine language

10. Which term means: “Second generation language using mnemonics”?

A. Assembly language
B. Imperative paradigm
C. Encapsulation
D. Object oriented paradigm
E. Fourth generation language

11. Which term means: “Third generation English-like portable language”?

A. Declarative paradigm
B. High-level language
C. Fifth generation language
D. Machine language
E. Class

12. Which term means: “Language where you state the result, such as SQL”?

A. Object oriented paradigm
B. Assembly language
C. Fourth generation language
D. Object
E. Programming paradigm

13. Which term means: “Logic and rule based language for AI such as Prolog”?

A. Class
B. Imperative paradigm
C. Encapsulation
D. High-level language
E. Fifth generation language

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

14. Which term means: “A style or approach to writing programs”?

A. Programming paradigm
B. Machine language
C. Object
D. Fourth generation language
E. Declarative paradigm

15. Which term means: “Style where statements change state step by step”?

A. Assembly language
B. Encapsulation
C. Fifth generation language
D. Imperative paradigm
E. Object oriented paradigm

16. Which term means: “Style that states what result is needed, not how”?

A. Programming paradigm
B. High-level language
C. Machine language
D. Declarative paradigm
E. Class

17. Which term means: “Style built on objects combining data and methods”?

A. Fourth generation language
B. Object
C. Imperative paradigm
D. Assembly language
E. Object oriented paradigm

18. Which term means: “Template that defines attributes and methods of objects”?

A. High-level language
B. Class
C. Fifth generation language
D. Encapsulation
E. Declarative paradigm

19. Which term means: “Instance of a class holding its own data”?

A. Programming paradigm
B. Fourth generation language
C. Object
D. Object oriented paradigm
E. Machine language

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

20. Which term means: “Keeping data and methods together and hiding details”?

A. Encapsulation
B. Assembly language
C. Class
D. Fifth generation language
E. Imperative paradigm

Structured Questions

1. Describe the five generations of programming languages. *(5 marks)*

> 1GL binary; 2GL assembly; 3GL high level; 4GL SQL; 5GL logic AI.

2. Compare imperative and declarative paradigms with examples. *(2 marks)*

> Imperative how, C/Python loop; declarative what, SQL/Prolog.

3. Explain class, object and encapsulation. *(3 marks)*

> Class template; object instance; encapsulation data + methods together.

4. Give two advantages of high-level languages over assembly. *(2 marks)*

> Easier to read/write; portable across machines.

5. Write Python to total a list in imperative and declarative style. *(2 marks)*

> for loop adding to total; sum(prices).

Summary

- Languages evolved from 1GL to 5GL.
- Higher generations are closer to people.
- Imperative says how.
- Declarative says what.
- OOP uses classes and objects.

MCQ answer key – 9.4

1. **C** – 1GL.
2. **A** – 2GL.
3. **E** – 4GL.
4. **B** – 5GL.
5. **C** – Declarative.
6. **D** – Class.
7. **D** – 120 + 80 + 200.

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

8. **B** – Encapsulation.
9. **E** – Machine language – First generation binary code run directly by the CPU.
10. **A** – Assembly language – Second generation language using mnemonics.
11. **B** – High-level language – Third generation English-like portable language.
12. **C** – Fourth generation language – Language where you state the result, such as SQL.
13. **E** – Fifth generation language – Logic and rule based language for AI such as Prolog.
14. **A** – Programming paradigm – A style or approach to writing programs.
15. **D** – Imperative paradigm – Style where statements change state step by step.
16. **D** – Declarative paradigm – Style that states what result is needed, not how.
17. **E** – Object oriented paradigm – Style built on objects combining data and methods.
18. **B** – Class – Template that defines attributes and methods of objects.
19. **C** – Object – Instance of a class holding its own data.
20. **A** – Encapsulation – Keeping data and methods together and hiding details.

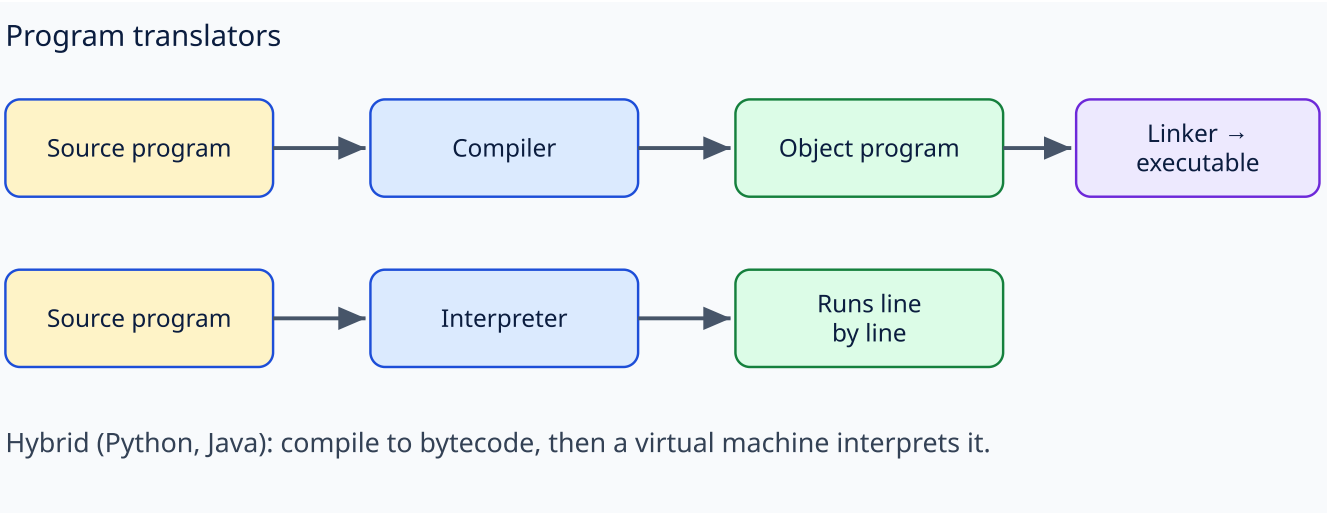
© 2026 schoolict.net (Ruwan Mangala Suraweera). All rights reserved.

Grade 13 A/L ICT · Lesson 9

9.5 Program Translation: Compilers, Interpreters and Linkers

Syllabus 9.5. A CPU understands only machine code, so a **source program** written in a high-level language must be **translated**. **Compilers** translate the whole program into an **object program**; **interpreters** translate and run one statement at a time; the **hybrid approach** compiles to bytecode that a virtual machine interprets. **Linkers** join object code and libraries into one executable.

1. Translators



Compiler, interpreter, linker and hybrid

Feature	Compiler	Interpreter
Translates	Whole program at once	One statement at a time
Output	Object/executable file	No separate file
Speed of running	Fast after compiling	Slower
Errors	Listed after translation	Stops at the first error
Examples	C, C++	JavaScript (classic), BASIC

2. Python uses the hybrid approach

🐍 Syntax error found before running vs runtime error

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

```
good = "x = 10\nprint(x * 2)"
bad = "print(10 +)"
code = compile(good, "good.py", "exec")   # translate to bytecode
exec(code)
try:
    compile(bad, "bad.py", "exec")
except SyntaxError:
    print("Syntax error: not translated")
try:
    print(10 / 0)
except ZeroDivisionError:
    print("Runtime error: division by zero")
```

▶ Output

```
20
Syntax error: not translated
Runtime error: division by zero
```

🧱 **Worked example: Why does the bad program never run, while the division error appears only when running?**
compile() translates source to **bytecode** first. "print(10 +)" breaks the grammar, so translation fails with a **syntax error** and nothing runs. 10 / 0 is grammatically correct, so it is translated, but fails at **run time**.

💡 Source program = what you write; object program = translated machine code.

🚀 **Beyond the syllabus (2026 update)** *Not required for the exam – read for interest.* In 2026 JavaScript engines and Python 3.13+ use just-in-time (JIT) compilation to speed up hot code while it runs.

Key Words

Source program – Program written by a programmer in a high-level language.

Object program – Machine code produced by translating source code.

Translator – Program that converts source code into machine code.

Compiler – Translates the whole program before running.

Interpreter – Translates and runs one statement at a time.

Hybrid approach – Compile to bytecode then interpret it on a virtual machine.

Bytecode – Intermediate code run by a virtual machine.

Linker – Joins object files and libraries into one executable.

Assembler – Translates assembly language into machine code.

Syntax error – Error in the grammar of the language found during translation.

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

Runtime error – Error that happens while the program is running.

Executable file – File that the operating system can run directly.

Sample Q&A

1. Why is translation needed?

> **Answer:** The CPU understands only machine code.

2. Compiler vs interpreter?

> **Answer:** Compiler translates all first; interpreter one statement at a time.

3. What is the hybrid approach?

> **Answer:** Compile to bytecode, then a virtual machine interprets it (Python, Java).

4. What does a linker do?

> **Answer:** Joins object files and libraries into an executable.

5. Which line of the translate program is never run?

> **Answer:** print(10 +) – it has a syntax error.

6. Source vs object program?

> **Answer:** Source is written by people; object is translated machine code.

MCQ

1. A translator that converts the whole program before running is a

A. compiler
B. loader
C. interpreter
D. linker
E. editor

2. A translator that runs one statement at a time is an

A. interpreter
B. assembler
C. debugger
D. linker
E. compiler

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

3. Joining object code with libraries is done by a

A. compiler
B. linker
C. editor
D. interpreter
E. debugger

4. Python and Java translate to bytecode first; this is the

A. pure compiler approach
B. assembly approach
C. linker approach
D. machine approach
E. hybrid approach

5. print(10 +) produces a

A. logic error
B. syntax error
C. runtime error
D. no error
E. linker error

6. 10 / 0 in Python produces a

A. runtime error
B. no error
C. syntax error
D. compile error
E. linker error

7. The program written by the programmer is the

A. bytecode only
B. source program
C. executable
D. object program
E. library

8. Assembly language is translated by an

A. browser
B. interpreter
C. assembler
D. editor
E. linker

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

9. Which term means: “Program written by a programmer in a high-level language”?

A. Translator
B. Linker
C. Hybrid approach
D. Source program
E. Runtime error

10. Which term means: “Machine code produced by translating source code”?

A. Assembler
B. Bytecode
C. Object program
D. Compiler
E. Executable file

11. Which term means: “Program that converts source code into machine code”?

A. Source program
B. Linker
C. Interpreter
D. Translator
E. Syntax error

12. Which term means: “Translates the whole program before running”?

A. Runtime error
B. Compiler
C. Object program
D. Hybrid approach
E. Assembler

13. Which term means: “Translates and runs one statement at a time”?

A. Executable file
B. Bytecode
C. Translator
D. Interpreter
E. Syntax error

14. Which term means: “Compile to bytecode then interpret it on a virtual machine”?

A. Runtime error
B. Source program
C. Compiler
D. Linker
E. Hybrid approach

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

15. Which term means: “Intermediate code run by a virtual machine”?

A. Object program
B. Interpreter
C. Bytecode
D. Assembler
E. Executable file

16. Which term means: “Joins object files and libraries into one executable”?

A. Source program
B. Hybrid approach
C. Linker
D. Syntax error
E. Translator

17. Which term means: “Translates assembly language into machine code”?

A. Bytecode
B. Object program
C. Compiler
D. Runtime error
E. Assembler

18. Which term means: “Error in the grammar of the language found during translation”?

A. Syntax error
B. Executable file
C. Translator
D. Interpreter
E. Linker

19. Which term means: “Error that happens while the program is running”?

A. Compiler
B. Assembler
C. Hybrid approach
D. Source program
E. Runtime error

20. Which term means: “File that the operating system can run directly”?

A. Bytecode
B. Object program
C. Syntax error
D. Executable file
E. Interpreter

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

Structured Questions

1. Explain why program translation is needed. *(2 marks)*

> CPU understands machine code only; high-level code must be converted.

2. Compare compilers and interpreters (three points). *(3 marks)*

> Whole vs line; object file vs none; error reporting/speed.

3. Describe the hybrid approach used by Python. *(3 marks)*

> Source to bytecode; virtual machine interprets; portable.

4. Explain the function of a linker. *(3 marks)*

> Combines object modules and libraries; resolves calls; makes executable.

5. Distinguish syntax and runtime errors with Python examples. *(2 marks)*

> print(10 +) syntax; 10 / 0 runtime.

Summary

- Source code must be translated to machine code.
- Compilers translate the whole program.
- Interpreters translate statement by statement.
- Python is hybrid: bytecode + virtual machine.
- Linkers build the final executable.

MCQ answer key – 9.5

1. **A** – Compiler.
2. **A** – Interpreter.
3. **B** – Linker.
4. **E** – Hybrid.
5. **B** – Grammar.
6. **A** – At run time.
7. **B** – Source.
8. **C** – Assembler.
9. **D** – Source program – Program written by a programmer in a high-level language.
10. **C** – Object program – Machine code produced by translating source code.
11. **D** – Translator – Program that converts source code into machine code.
12. **B** – Compiler – Translates the whole program before running.
13. **D** – Interpreter – Translates and runs one statement at a time.
14. **E** – Hybrid approach – Compile to bytecode then interpret it on a virtual machine.
15. **C** – Bytecode – Intermediate code run by a virtual machine.

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

16. **C** – Linker – Joins object files and libraries into one executable.
17. **E** – Assembler – Translates assembly language into machine code.
18. **A** – Syntax error – Error in the grammar of the language found during translation.
19. **E** – Runtime error – Error that happens while the program is running.
20. **D** – Executable file – File that the operating system can run directly.

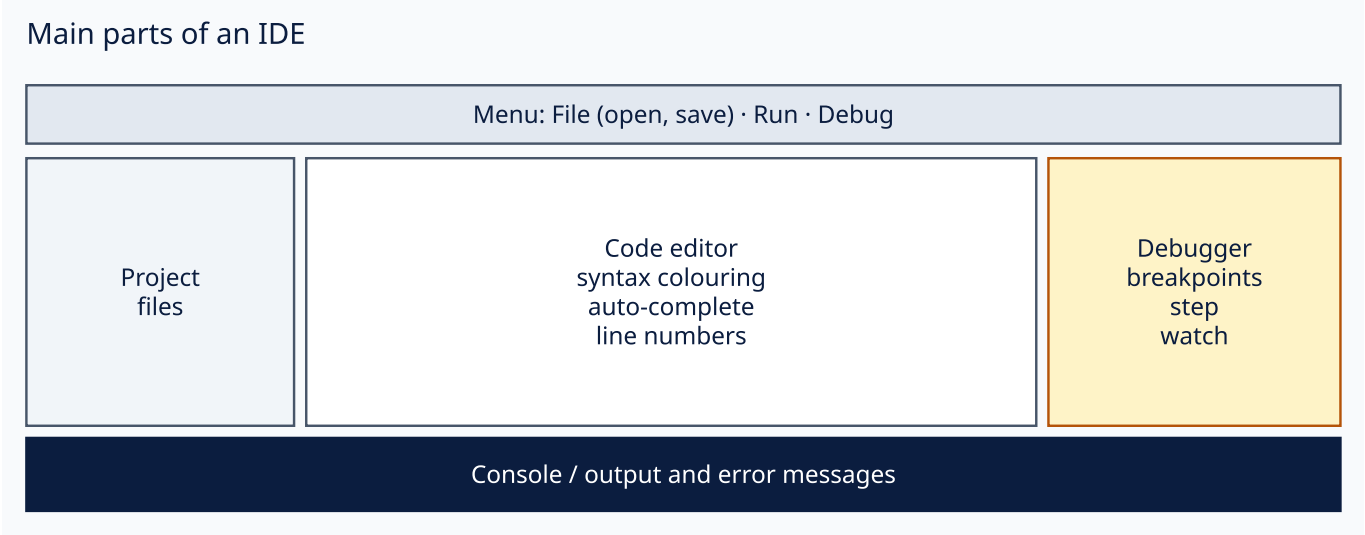
© 2026 schoolict.net (Ruwan Mangala Suraweera). All rights reserved.

Grade 13 A/L ICT · Lesson 9

9.6 Integrated Development Environments and Debugging

Syllabus 9.6. An **IDE** brings together an editor, translator, runner and **debugger** in one program (for example IDLE, Thonny or VS Code). You learn to **open and save files**, **run** programs and use **debugging facilities** such as breakpoints, stepping and watching variables.

1. Basic features of an IDE



Parts of an IDE

- **Editor** with syntax colouring, auto-indent, auto-complete and line numbers.
- **File menu**: New, Open, Save, Save As (Python files end in .py).
- **Run**: translate and execute (F5 in IDLE); output and errors appear in the **console**.
- **Debugger**: breakpoints, step into/over, watch variables and the call stack.

2. Debugging an off-by-one error

🐍 Debugging an off-by-one error

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

```
marks = [60, 70, 80]

def average_bug(m):
    total = 0
    for i in range(len(m) - 1):   # bug: skips the last item
        total += m[i]
    return total / len(m)

def average_fixed(m):
    total = 0
    for i in range(len(m)):
        total += m[i]
    return total / len(m)

print("Buggy:", average_bug(marks))
print("Fixed:", average_fixed(marks))
```

▶ Output

```
Buggy: 43.333333333333336
Fixed: 70.0
```

🧱 **Worked example: The buggy average of 60, 70, 80 is wrong. How does a debugger find the cause?**
Set a breakpoint inside the loop and watch i and total. i takes only 0 and 1, so total reaches 130 and 130 ÷ 3 = **43.33**. The range should be range(len(m)); then total = 210 and the average is **70.0**.

Error type	Found by	Example
Syntax	Translator, before running	missing colon after if
Runtime	Program crash while running	division by zero
Logic	Wrong output; use the debugger	range(len(m) - 1)

💡 Logic errors give no error message – only testing with expected results reveals them.

🚀 **Beyond the syllabus (2026 update)** *Not required for the exam – read for interest.* In 2026 IDEs include AI assistants that explain errors and suggest fixes; always test the suggestion before accepting it.

Key Words

IDE – Software combining editor, translator, runner and debugger.

Code editor – Part of an IDE where source code is typed.

Syntax highlighting – Colouring keywords and strings to aid reading.

Auto-complete – Suggesting names while typing.

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

Console – Window showing program output and errors.	**Debugger** – Tool for running a program step by step to find errors.
Breakpoint – Marked line where execution pauses.	**Step into** – Run the next line, entering any function called.
Watch – Showing the current value of chosen variables.	**Logic error** – Mistake that gives wrong results without crashing.
Off-by-one error – Loop that runs one time too many or too few.	**Call stack** – List of functions currently being executed.

Sample Q&A

1. Name four IDE features.

> **Answer:** Editor, file open/save, run, debugger.

2. What is a breakpoint?

> **Answer:** A line where execution pauses so you can inspect variables.

3. Buggy and fixed averages?

> **Answer:** 43.33 and 70.0.

4. What caused the bug?

> **Answer:** range(len(m) - 1) skipped the last item.

5. Which errors give no message?

> **Answer:** Logic errors.

6. How do you save a Python file?

> **Answer:** File → Save / Save As with a .py name.

MCQ

1. The IDE part that pauses at a marked line is a

A. linker
B. breakpoint
C. console
D. comment
E. compiler

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

2. Showing a variable value during debugging is

A. compile all
B. comment out
C. save as
D. link
E. watch

3. The buggy program prints average

A. 80.00
B. 60.00
C. 70.00
D. 43.33
E. 65.00

4. The fixed program prints average

A. 65.0
B. 210.0
C. 70.0
D. 80.0
E. 43.3

5. range(len(m) - 1) instead of range(len(m)) is an

A. off-by-one logic error
B. linker error
C. syntax error
D. runtime crash
E. hardware error

6. Output and error messages appear in the

A. structure chart
B. breakpoint
C. console
D. file menu
E. linker

7. A missing colon after if is found

A. before the program runs (syntax error)
B. only after output
C. by the printer
D. by the linker
E. never

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

8. Python source files are saved with the extension

A. .exe
B. .docx
C. .obj
D. .txt only
E. .py

9. Which term means: “Software combining editor, translator, runner and debugger”?

A. Off-by-one error
B. Step into
C. Debugger
D. IDE
E. Syntax highlighting

10. Which term means: “Part of an IDE where source code is typed”?

A. Auto-complete
B. Breakpoint
C. Watch
D. Call stack
E. Code editor

11. Which term means: “Colouring keywords and strings to aid reading”?

A. Step into
B. Syntax highlighting
C. IDE
D. Logic error
E. Console

12. Which term means: “Suggesting names while typing”?

A. Auto-complete
B. Debugger
C. Code editor
D. Watch
E. Off-by-one error

13. Which term means: “Window showing program output and errors”?

A. Logic error
B. Console
C. Syntax highlighting
D. Breakpoint
E. Call stack

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

14. Which term means: “Tool for running a program step by step to find errors”?

A. IDE
B. Off-by-one error
C. Debugger
D. Auto-complete
E. Step into

15. Which term means: “Marked line where execution pauses”?

A. Call stack
B. Console
C. Watch
D. Breakpoint
E. Code editor

16. Which term means: “Run the next line, entering any function called”?

A. Step into
B. Debugger
C. Syntax highlighting
D. IDE
E. Logic error

17. Which term means: “Showing the current value of chosen variables”?

A. Off-by-one error
B. Auto-complete
C. Watch
D. Code editor
E. Breakpoint

18. Which term means: “Mistake that gives wrong results without crashing”?

A. Step into
B. Logic error
C. Call stack
D. Syntax highlighting
E. Console

19. Which term means: “Loop that runs one time too many or too few”?

A. Debugger
B. Auto-complete
C. Watch
D. Off-by-one error
E. IDE

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

20. Which term means: “List of functions currently being executed”?

A. Console
B. Logic error
C. Code editor
D. Breakpoint
E. Call stack

Structured Questions

1. Describe four basic features of an IDE. *(4 marks)*

> Editor; file management; run/console; debugger.

2. List the steps to open, edit, save and run a Python file in an IDE. *(5 marks)*

> File Open; edit; File Save as .py; Run (F5); read console.

3. Explain breakpoint, step and watch. *(3 marks)*

> Pause at line; run one line; show variable values.

4. Find and fix the bug in average_bug(). *(3 marks)*

> range(len(m) - 1) skips last; use range(len(m)); 70.0.

5. Distinguish syntax, runtime and logic errors. *(3 marks)*

> Syntax grammar; runtime crash; logic wrong output.

Summary

- An IDE combines editor, runner and debugger.
- Save Python files as .py.
- Breakpoints pause execution.
- Step and watch reveal logic errors.
- Logic errors need testing to be found.

MCQ answer key – 9.6

1. **B** – Breakpoint.
2. **E** – Watch.
3. **D** – 130 ÷ 3.
4. **C** – 210 ÷ 3.
5. **A** – Logic.
6. **C** – Console.
7. **A** – Syntax.

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

8. **E** – .py.
9. **D** – IDE – Software combining editor, translator, runner and debugger.
10. **E** – Code editor – Part of an IDE where source code is typed.
11. **B** – Syntax highlighting – Colouring keywords and strings to aid reading.
12. **A** – Auto-complete – Suggesting names while typing.
13. **B** – Console – Window showing program output and errors.
14. **C** – Debugger – Tool for running a program step by step to find errors.
15. **D** – Breakpoint – Marked line where execution pauses.
16. **A** – Step into – Run the next line, entering any function called.
17. **C** – Watch – Showing the current value of chosen variables.
18. **B** – Logic error – Mistake that gives wrong results without crashing.
19. **D** – Off-by-one error – Loop that runs one time too many or too few.
20. **E** – Call stack – List of functions currently being executed.

© 2026 schoolict.net (Ruwan Mangala Suraweera). All rights reserved.

Grade 13 A/L ICT · Lesson 9

9.7 Python Basics: Structure, Variables, Data Types, Operators and I/O

Syllabus 9.7. Here we use an imperative language – Python – to encode algorithms. You meet the **structure** of a program, **comments**, **constants and variables**, the **primitive data types**, **arithmetic, relational, logical and bitwise operators**, **operator precedence**, keyboard **input** and screen **output**.

1. Structure, comments, constants and variables

🐍 Constants, variables, types and input

```
RATE = 0.08                      # constant (by convention UPPER CASE)
name = input("Name: ")           # str
qty = int(input("Quantity: "))   # int
price = 250.0                    # float
paid = True                      # bool
total = qty * price * (1 + RATE)
print(name, "pays Rs.", round(total, 2))
print(type(qty).__name__, type(price).__name__, type(paid).__name__)
```

▶ Output

```
Name: Nimal
Quantity: 4
Nimal pays Rs. 1080.0
int float bool
```

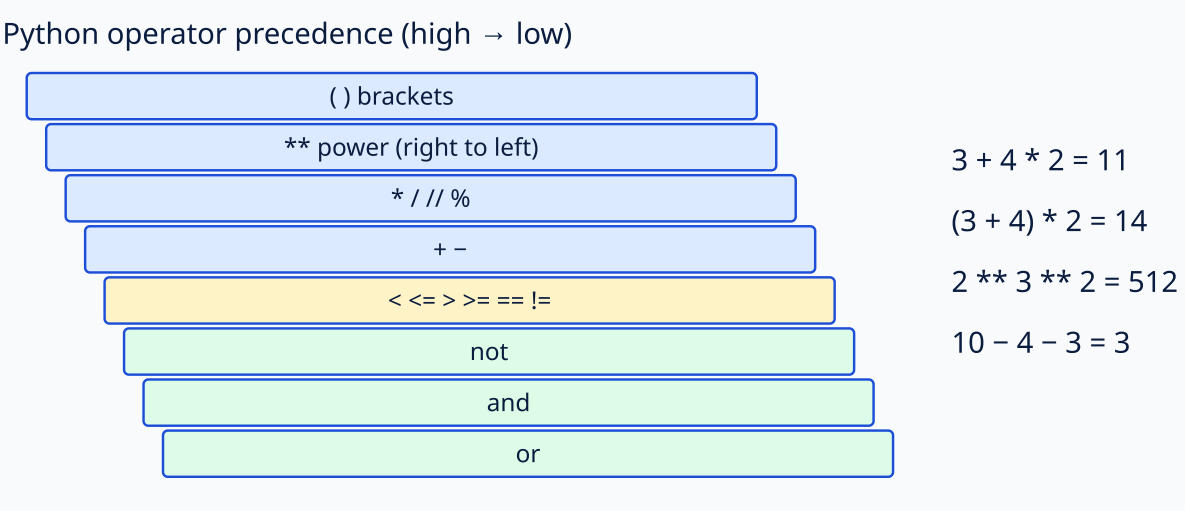
Type	Example	Meaning
int	4	Whole number
float	250.0	Real (decimal) number
str	"Nimal"	Text
bool	True / False	Truth value

A **comment** starts with # and is ignored by the translator. A **variable** is a named memory location whose value can change; a **constant** (written in UPPER CASE by convention) should not change. input() always returns a **str**, so int() or float() converts it.

> 🧱 **Worked example: Nimal buys 4 items at Rs. 250.0 with 8% tax. What is printed?**
> 4 × 250.0 = 1000.0; × 1.08 = **1080.0**, so it prints “Nimal pays Rs. 1080.0”. type() shows int, float, bool.

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

2. Operators and precedence



Operator precedence

Category	Operators	Example → result
Arithmetic	+ - * / // % **	7 // 2 → 3, 7 % 2 → 1, 7 / 2 → 3.5
Relational	== != < <= > >=	5 > 3 → True
Logical	and or not	5 > 3 and 2 > 8 → False
Bitwise	& | ^ ~ << >>	5 & 3 → 1, 5 | 3 → 7, 5 ^ 3 → 6, 5 << 1 → 10

🐍 Operators and precedence

```
print(7 // 2, 7 % 2, 7 / 2)     # arithmetic
print(3 + 4 * 2, (3 + 4) * 2)  # precedence
print(2 ** 3 ** 2)             # ** is right to left
print(10 - 4 - 3)              # - is left to right
print(5 > 3 and 2 > 8)         # relational + logical
print(not True or True)
print(5 & 3, 5 | 3, 5 ^ 3, 5 << 1)   # bitwise
```

▶ Output

```
3 1 3.5
11 14
512
3
False
True
1 7 6 10
```

🧱 **Worked example: Explain 2 ** 3 ** 2 and 5 ^ 3.**

groups right to left**: 3** 2 = 9, then 2 **9 =** 512**. 5 = 101 and 3 = 011 in binary; XOR gives 110 =** 6**.

💡 In Python ^ is bitwise XOR, not power – use ** for powers.

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

🚀 **Beyond the syllabus (2026 update)** *Not required for the exam – read for interest.* Python 3.12+ in 2026 gives clearer error messages and optional type hints (qty: int) that IDEs check before running.

Key Words

Comment – Note in code starting with # that is not executed.

Variable – Named storage location whose value can change.

Constant – Named value that should not change while running.

Primitive data type – Basic built-in type such as int, float, str or bool.

Integer – Whole number type int.

Float – Number type with a decimal part.

Boolean – Type holding True or False.

Arithmetic operator – Operator such as + or // that calculates numbers.

Relational operator – Operator that compares values and gives True or False.

Logical operator – and, or, not combining conditions.

Bitwise operator – Operator working on individual bits such as & or ^.

Operator precedence – Order in which operators are evaluated.

Sample Q&A

1. What does input() return?

> **Answer:** A string (str).

2. 7 // 2 and 7 % 2?

> **Answer:** 3 and 1.

3. 3 + 4 * 2?

> **Answer:** 11, because * is done before +.

4. Value of 10 - 4 - 3?

> **Answer:** 3 (left to right).

5. Four primitive types?

> **Answer:** int, float, str, bool.

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

6. Constant vs variable?

> **Answer:** A constant should not change; a variable can.

MCQ

1. 7 // 2 in Python gives

A. 1
B. 3.5
C. 4
D. 3
E. 2

2. 7 % 2 in Python gives

A. 1
B. 0
C. 2
D. 3.5
E. 3

3. 3 + 4 * 2 gives

A. 14
B. 11
C. 9
D. 10
E. 24

4. 2 3 2 gives

A. 64
B. 12
C. 512
D. 256
E. 36

5. 5 ^ 3 in Python gives

A. 6
B. 2
C. 8
D. 125
E. 15

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

6. not True or True gives

A. 0
B. error
C. False
D. True
E. None

7. The type of the value returned by input() is

A. bool
B. float
C. int
D. list
E. str

8. Nimal’s total with 8% tax on 4 × 250.0 is

A. 1008.0
B. 1000.0
C. 1080.0
D. 1250.0
E. 1800.0

9. Which term means: “Note in code starting with # that is not executed”?

A. Bitwise operator
B. Comment
C. Arithmetic operator
D. Float
E. Constant

10. Which term means: “Named storage location whose value can change”?

A. Operator precedence
B. Relational operator
C. Variable
D. Primitive data type
E. Boolean

11. Which term means: “Named value that should not change while running”?

A. Integer
B. Logical operator
C. Arithmetic operator
D. Constant
E. Comment

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

12. Which term means: “Basic built-in type such as int, float, str or bool”?

A. Bitwise operator
B. Variable
C. Relational operator
D. Float
E. Primitive data type

13. Which term means: “Whole number type int”?

A. Logical operator
B. Integer
C. Operator precedence
D. Boolean
E. Constant

14. Which term means: “Number type with a decimal part”?

A. Bitwise operator
B. Comment
C. Primitive data type
D. Arithmetic operator
E. Float

15. Which term means: “Type holding True or False”?

A. Operator precedence
B. Boolean
C. Relational operator
D. Integer
E. Variable

16. Which term means: “Operator such as + or // that calculates numbers”?

A. Arithmetic operator
B. Constant
C. Logical operator
D. Comment
E. Float

17. Which term means: “Operator that compares values and gives True or False”?

A. Bitwise operator
B. Boolean
C. Primitive data type
D. Relational operator
E. Variable

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

18. Which term means: “and, or, not combining conditions”?

A. Logical operator
B. Constant
C. Integer
D. Operator precedence
E. Arithmetic operator

19. Which term means: “Operator working on individual bits such as & or ^”?

A. Comment
B. Float
C. Bitwise operator
D. Relational operator
E. Primitive data type

20. Which term means: “Order in which operators are evaluated”?

A. Logical operator
B. Integer
C. Variable
D. Boolean
E. Operator precedence

Structured Questions

1. Name four primitive data types with examples. *(4 marks)*

> int 4; float 250.0; str "Nimal"; bool True.

2. Explain variable, constant and comment with code. *(3 marks)*

> qty = 4 variable; RATE = 0.08 constant; # comment ignored.

3. Evaluate 3 + 4 * 2, (3 + 4) * 2 and 2 ** 3 ** 2 and explain. *(3 marks)*

> 11 (* first); 14 (brackets); 512 (** right to left).

4. Give one example each of relational, logical and bitwise operators with results. *(3 marks)*

> 5 > 3 True; 5 > 3 and 2 > 8 False; 5 & 3 = 1.

5. Write a program that inputs quantity and prints the total with 8% tax. *(3 marks)*

> int(input()); total = qty * price * 1.08; print round.

Summary

- Comments start with #.

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

- int, float, str and bool are primitive types.
- input() returns a string – convert it.
- ** before * / // % before + −.
- ^ is XOR; ** is power.

MCQ answer key – 9.7

1. **D** – Floor division.
2. **A** – Remainder.
3. **B** – * first.
4. **C** – Right to left.
5. **A** – XOR.
6. **D** – not first.
7. **E** – Always str.
8. **C** – 1000 × 1.08.
9. **B** – Comment – Note in code starting with # that is not executed.
10. **C** – Variable – Named storage location whose value can change.
11. **D** – Constant – Named value that should not change while running.
12. **E** – Primitive data type – Basic built-in type such as int, float, str or bool.
13. **B** – Integer – Whole number type int.
14. **E** – Float – Number type with a decimal part.
15. **B** – Boolean – Type holding True or False.
16. **A** – Arithmetic operator – Operator such as + or // that calculates numbers.
17. **D** – Relational operator – Operator that compares values and gives True or False.
18. **A** – Logical operator – and, or, not combining conditions.
19. **C** – Bitwise operator – Operator working on individual bits such as & or ^.
20. **E** – Operator precedence – Order in which operators are evaluated.

© 2026 schoolict.net (Ruwan Mangala Suraweera). All rights reserved.

Grade 13 A/L ICT · Lesson 9

9.8 Control Structures: Sequence, Selection and Repetition

Syllabus 9.8. Every algorithm is built from three **control structures**: **sequence** (steps in order), **selection** (if / elif / else) and **repetition** – **iteration** with a known count (for) and **looping** while a condition is true (while). Structures can be **nested** inside each other.

1. The three structures

Sequence
Statements run one after another.

Selection
if, elif, else choose a path using a condition.

Repetition
for repeats a set number of times; while repeats while a condition is True.

🐍 Sequence, selection and repetition

```
mark = 67
if mark >= 75:
    print("A")
elif mark >= 65:
    print("B")
elif mark >= 50:
    print("C")
else:
    print("F")

for i in range(1, 4):        # iteration (count-controlled)
    print(i, end=" ")
print()

n = 472                      # looping (condition-controlled)
digit_sum = 0
while n > 0:
    digit_sum += n % 10
    n = n // 10
print("Digit sum:", digit_sum)

for r in range(1, 4):        # nested loops
    for c in range(1, 4):
        print(r * c, end=" ")
    print()
```

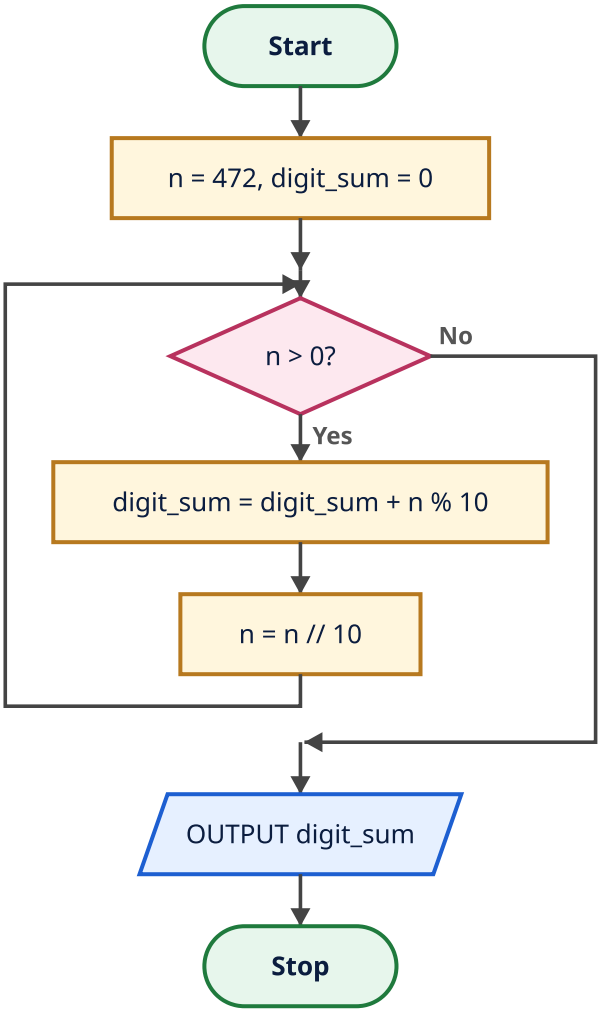
▶ Output

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

```
B
1 2 3
Digit sum: 13
1 2 3
2 4 6
3 6 9
```

🧱 **Worked example: Why does mark 67 print B?**
Conditions are tested in order: 67 ≥ 75 is False, 67 ≥ 65 is **True**, so B is printed and the rest are skipped.

2. Tracing the while loop



Flowchart: sum of digits of 472

n % 10	digit_sum	n after // 10
2	2	47
7	9	4
4	13	0

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

🧮 **Worked example: Trace the digit-sum loop for 472.**
The loop runs **3** times: 2 → sum 2, 7 → sum 9, 4 → sum 13. n becomes 0, the condition is False and it prints **Digit sum: 13**.

💡 Use for when you know how many times; use while when you stop on a condition. Indentation shows what is inside.

🚀 **Beyond the syllabus (2026 update)** *Not required for the exam – read for interest.* Python 3.10+ adds match-case, a neat multi-way selection that many 2026 programs use instead of long elif chains.

Key Words

Control structure – Way of controlling the order in which statements run.	**Sequence** – Statements executed one after another.
Selection – Choosing a path depending on a condition.	**Repetition** – Running a block of statements more than once.
Iteration – Count-controlled repetition such as a for loop.	**Looping** – Condition-controlled repetition such as a while loop.
if statement – Runs a block only when its condition is True.	**elif** – Further condition tested when earlier ones are False.
for loop – Repeats for each value in a range or sequence.	**while loop** – Repeats while a condition stays True.
Nested structure – A control structure placed inside another.	**Infinite loop** – Loop whose condition never becomes False.

Sample Q&A

1. Three control structures?

Answer: Sequence, selection, repetition.

2. Iteration vs looping?

Answer: Iteration is count-controlled (for); looping is condition-controlled (while).

3. What does range(1, 4) give?

Answer: 1, 2, 3.

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

4. Digit sum of 472?

> **Answer:** 13.

5. Last row of the 3 × 3 table?

> **Answer:** 3 6 9.

6. What causes an infinite loop?

> **Answer:** A condition that never becomes False, e.g. forgetting n = n // 10.

MCQ

1. if / elif / else is an example of

A. iteration
B. recursion
C. looping
D. selection
E. sequence

2. A for loop over range(1, 4) is

A. iteration (count-controlled)
B. sequence only
C. selection
D. looping (condition-controlled)
E. a function

3. Mark 67 in the program prints

A. nothing
B. A
C. F
D. B
E. C

4. range(1, 4) produces

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

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

5. The digit sum of 472 printed is

A. 14
B. 472
C. 7
D. 11
E. 13

6. The while loop body runs for 472

A. 472 times
B. 2 times
C. 1 time
D. 3 times
E. 4 times

7. Placing a loop inside another loop is

A. compiling
B. nesting
C. sequencing
D. linking
E. tracing

8. The middle row of the nested 3 × 3 table is

A. 1 2 3
B. 4 5 6
C. 3 6 9
D. 2 4 6
E. 2 3 4

9. Which term means: “Way of controlling the order in which statements run”?

A. Looping
B. Selection
C. Control structure
D. Nested structure
E. elif

10. Which term means: “Statements executed one after another”?

A. Infinite loop
B. Sequence
C. Repetition
D. for loop
E. if statement

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

11. Which term means: “Choosing a path depending on a condition”?

A. while loop
B. elif
C. Selection
D. Iteration
E. Control structure

12. Which term means: “Running a block of statements more than once”?

A. Repetition
B. Nested structure
C. Sequence
D. Looping
E. for loop

13. Which term means: “Count-controlled repetition such as a for loop”?

A. Selection
B. Iteration
C. while loop
D. if statement
E. Infinite loop

14. Which term means: “Condition-controlled repetition such as a while loop”?

A. Looping
B. Repetition
C. elif
D. Control structure
E. Nested structure

15. Which term means: “Runs a block only when its condition is True”?

A. Infinite loop
B. for loop
C. if statement
D. Sequence
E. Iteration

16. Which term means: “Further condition tested when earlier ones are False”?

A. while loop
B. Selection
C. Control structure
D. Looping
E. elif

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

17. Which term means: “Repeats for each value in a range or sequence”?

A. Sequence
B. Nested structure
C. for loop
D. if statement
E. Repetition

18. Which term means: “Repeats while a condition stays True”?

A. elif
B. while loop
C. Selection
D. Infinite loop
E. Iteration

19. Which term means: “A control structure placed inside another”?

A. Nested structure
B. Control structure
C. Looping
D. Repetition
E. for loop

20. Which term means: “Loop whose condition never becomes False”?

A. Sequence
B. if statement
C. while loop
D. Iteration
E. Infinite loop

Structured Questions

1. Describe the three control structures with Python examples. *(3 marks)*

> Sequence statements; selection if/elif/else; repetition for/while.

2. Write a program that prints A, B, C or F for a mark. *(4 marks)*

> if >= 75 A; elif >= 65 B; elif >= 50 C; else F.

3. Trace the digit-sum loop for 472. *(4 marks)*

> 2/2/47; 7/9/4; 4/13/0; output 13.

4. Compare for and while loops. *(3 marks)*

> for known count; while condition; both repeat a block.

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

5. Write nested loops to print a 3 × 3 multiplication table. *(4 marks)*

> for r in range(1, 4); for c in range(1, 4); print(r * c, end); print().

Summary

- Sequence, selection and repetition build every algorithm.
- if/elif/else tests conditions in order.
- for = iteration with a known count.
- while = looping until a condition is False.
- Structures can be nested.

MCQ answer key – 9.8

1. **D** – Selection.
2. **A** – Iteration.
3. **D** – 67 ≥ 65.
4. **E** – Stop excluded.
5. **E** – 4 + 7 + 2.
6. **D** – Three digits.
7. **B** – Nested.
8. **D** – 2 × 1..3.
9. **C** – Control structure – Way of controlling the order in which statements run.
10. **B** – Sequence – Statements executed one after another.
11. **C** – Selection – Choosing a path depending on a condition.
12. **A** – Repetition – Running a block of statements more than once.
13. **B** – Iteration – Count-controlled repetition such as a for loop.
14. **A** – Looping – Condition-controlled repetition such as a while loop.
15. **C** – if statement – Runs a block only when its condition is True.
16. **E** – elif – Further condition tested when earlier ones are False.
17. **C** – for loop – Repeats for each value in a range or sequence.
18. **B** – while loop – Repeats while a condition stays True.
19. **A** – Nested structure – A control structure placed inside another.
20. **E** – Infinite loop – Loop whose condition never becomes False.

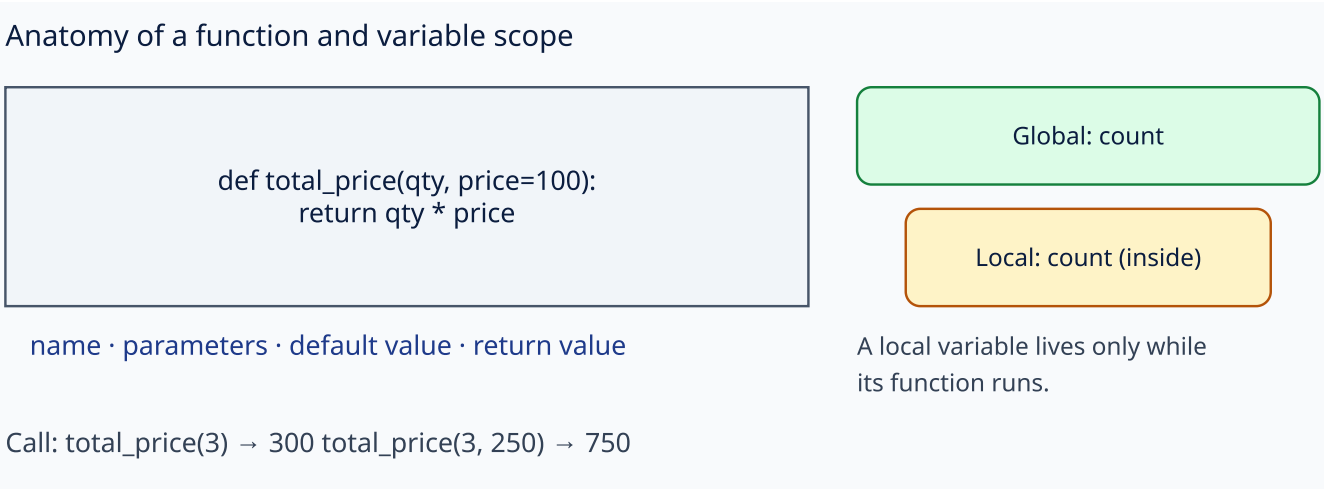
© 2026 schoolict.net (Ruwan Mangala Suraweera). All rights reserved.

Grade 13 A/L ICT · Lesson 9

9.9 Functions: Built-in and User-defined

Syllabus 9.9. A **sub-program** (function) is a named block that does one task. Python has **built-in** functions (len, max, round) and you can write **user-defined** functions with **parameters**, **return values** and **default values**. The **scope** of a variable decides where it can be used: **local** variables live only inside their function; **global** variables live for the whole program.

1. Structure of a function



Parts of a function and scope

- def name(parameters): starts the function; the indented body follows.
- **Parameters** receive values; the values passed in a call are **arguments**.
- **return** sends a value back to the caller; without it the function returns None.
- A **default value** (price=100) is used when that argument is left out.

🐍 Parameters, return values, defaults and scope

```
count = 0                          # global variable

def total_price(qty, price=100):   # price has a default value
    return qty * price             # return value

def add_visit():
    global count
    count += 1

def show_local():
    count = 99                     # local: hides the global
    return count

print(total_price(3))
print(total_price(3, 250))
add_visit(); add_visit()
print("Local:", show_local(), "Global:", count)
print(len("Lanka"), max(4, 9, 2), round(3.567, 1))   # built-in
```

▶ Output

```
300
750
Local: 99 Global: 2
5 9 3.6
```

🧱 **Worked example: Explain each printed line of the functions program.**
total_price(3) uses the default 100 → **300**; total_price(3, 250) → **750**. add_visit() runs twice and changes the **global** count to 2. show_local() makes a **local** count = 99 that disappears when the function ends, so it prints “Local: 99 Global: 2”. Built-ins give len("Lanka") = 5, max = 9, round(3.567, 1) = 3.6.

Local variable	Global variable
Created inside a function	Created outside all functions
Lives only while the function runs	Lives until the program ends
Used only in that function	Readable everywhere; change it inside with global

💡 Prefer parameters and return values to global variables – functions become easier to test and reuse.

🚀 **Beyond the syllabus (2026 update)** *Not required for the exam – read for interest.* Type hints such as def total_price(qty: int, price: float = 100) -> float are common in 2026 code and help IDEs catch mistakes.

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

Key Words

Sub-program – Named block of code that performs a task.

Built-in function – Function provided by Python such as len or max.

User-defined function – Function written by the programmer with def.

Parameter – Variable in a function definition that receives a value.

Argument – Actual value passed to a function in a call.

Return value – Value sent back to the caller by return.

Default value – Value a parameter takes when no argument is given.

Scope – Part of a program where a variable can be used.

Local variable – Variable that exists only inside its function.

Global variable – Variable defined outside functions for the whole program.

Lifetime – How long a variable exists in memory.

Function call – Statement that runs a function by its name.

Sample Q&A

1. Built-in vs user-defined?

> **Answer:** Built-in come with Python; user-defined are written with def.

2. Parameter vs argument?

> **Answer:** Parameter is in the definition; argument is the value passed.

3. total_price(3) returns?

> **Answer:** 300 (default price 100).

4. Why does Global stay 2?

> **Answer:** show_local() changes only its own local count.

5. Lifetime of a local variable?

> **Answer:** Only while the function runs.

6. Why use return values?

> **Answer:** To send results back so the caller can use them.

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

MCQ

1. len, max and round are

A. modules only
B. keywords
C. built-in functions
D. user-defined functions
E. constants

2. The value passed in total_price(3, 250) for price is an

A. argument
B. global
C. return value
D. parameter name
E. default value

3. total_price(3) returns

A. 100
B. None
C. 300
D. 3
E. 750

4. total_price(3, 250) returns

A. None
B. 750
C. 253
D. 300
E. 250

5. After two add_visit() calls the global count is

A. 99
B. 0
C. 1
D. 3
E. 2

6. show_local() returns

A. 2
B. None
C. 99
D. 1
E. 0

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

7. A variable that exists only while its function runs is

A. global
B. constant
C. built-in
D. local
E. default

8. round(3.567, 1) gives

A. 3.57
B. 3.0
C. 4
D. 3.5
E. 3.6

9. Which term means: “Named block of code that performs a task”?

A. Lifetime
B. Scope
C. Return value
D. Sub-program
E. User-defined function

10. Which term means: “Function provided by Python such as len or max”?

A. Default value
B. Built-in function
C. Local variable
D. Parameter
E. Function call

11. Which term means: “Function written by the programmer with def”?

A. User-defined function
B. Global variable
C. Argument
D. Scope
E. Sub-program

12. Which term means: “Variable in a function definition that receives a value”?

A. Lifetime
B. Return value
C. Built-in function
D. Parameter
E. Local variable

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

13. Which term means: “Actual value passed to a function in a call”?

A. Global variable
B. User-defined function
C. Argument
D. Function call
E. Default value

14. Which term means: “Value sent back to the caller by return”?

A. Parameter
B. Return value
C. Sub-program
D. Scope
E. Lifetime

15. Which term means: “Value a parameter takes when no argument is given”?

A. Default value
B. Function call
C. Local variable
D. Argument
E. Built-in function

16. Which term means: “Part of a program where a variable can be used”?

A. Scope
B. Global variable
C. Sub-program
D. Return value
E. User-defined function

17. Which term means: “Variable that exists only inside its function”?

A. Parameter
B. Local variable
C. Lifetime
D. Built-in function
E. Default value

18. Which term means: “Variable defined outside functions for the whole program”?

A. User-defined function
B. Function call
C. Scope
D. Global variable
E. Argument

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

19. Which term means: “How long a variable exists in memory”?

A. Local variable
B. Sub-program
C. Return value
D. Parameter
E. Lifetime

20. Which term means: “Statement that runs a function by its name”?

A. Default value
B. Argument
C. Built-in function
D. Global variable
E. Function call

Structured Questions

1. Explain built-in and user-defined functions with examples. *(3 marks)*

> Built-in len/max; user-defined with def; example.

2. Label the parts of def total_price(qty, price=100). *(5 marks)*

> def keyword; name; parameters; default value; return.

3. Compare local and global variables (three points). *(3 marks)*

> Where created; lifetime; where usable.

4. Write a function area(l, w=1) that returns l × w and call it twice. *(3 marks)*

> def area(l, w=1): return l * w; area(5) → 5; area(5, 3) → 15.

5. Explain why show_local() does not change the global count. *(3 marks)*

> Assignment makes a local; global unchanged; use global keyword to change.

Summary

- Functions are named reusable blocks.
- Parameters receive arguments.
- return sends a value back.
- Default values fill missing arguments.
- Local variables live only inside the function.

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

MCQ answer key – 9.9

1. **C** – Built-in.
2. **A** – Argument.
3. **C** – 3 × 100.
4. **B** – 3 × 250.
5. **E** – Two increments.
6. **C** – Local value.
7. **D** – Local.
8. **E** – One decimal.
9. **D** – Sub-program – Named block of code that performs a task.
10. **B** – Built-in function – Function provided by Python such as len or max.
11. **A** – User-defined function – Function written by the programmer with def.
12. **D** – Parameter – Variable in a function definition that receives a value.
13. **C** – Argument – Actual value passed to a function in a call.
14. **B** – Return value – Value sent back to the caller by return.
15. **A** – Default value – Value a parameter takes when no argument is given.
16. **A** – Scope – Part of a program where a variable can be used.
17. **B** – Local variable – Variable that exists only inside its function.
18. **D** – Global variable – Variable defined outside functions for the whole program.
19. **E** – Lifetime – How long a variable exists in memory.
20. **E** – Function call – Statement that runs a function by its name.

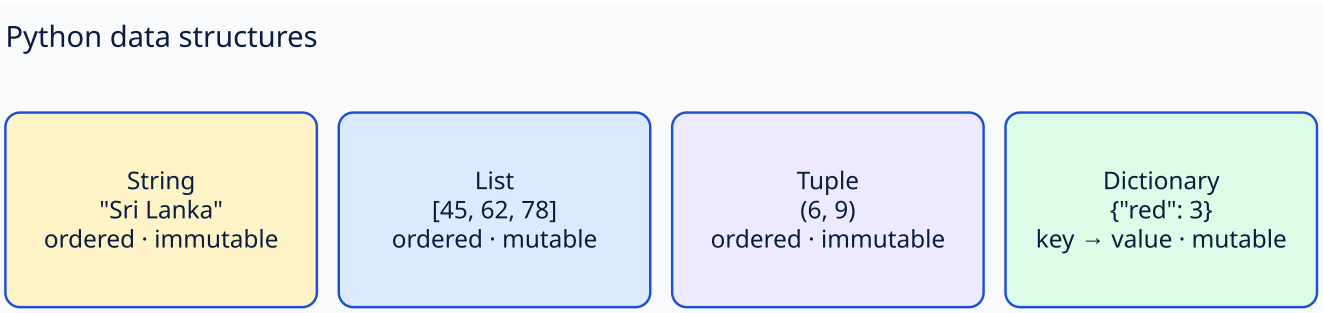
© 2026 schoolict.net (Ruwan Mangala Suraweera). All rights reserved.

Grade 13 A/L ICT · Lesson 9

9.10 Data Structures: Strings, Lists, Tuples and Dictionaries

Syllabus 9.10. A **data structure** stores many values under one name. Python has **strings** (text), **lists** (changeable sequences), **tuples** (fixed sequences) and **dictionaries** (key → value pairs). You use **indexing**, **slicing** and **methods** to work with them.

1. Four structures



String, list, tuple and dictionary

Structure	Brackets	Ordered?	Changeable?
String	" "	Yes	No
List	[ ]	Yes	Yes
Tuple	( )	Yes	No
Dictionary	{key: value}	By insertion	Yes

🐍 Strings, lists, tuples and dictionaries

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

```
s = "Sri Lanka"
print(s[0], s[-1], s[4:9], len(s), s.upper())

marks = [45, 78, 62]
marks.append(90)
marks.sort()
print(marks, marks[1:3], sum(marks))

point = (6, 9)                 # tuple: cannot be changed
print(point[0] + point[1])

words = "red blue red green red".split()
freq = {}
for w in words:
    freq[w] = freq.get(w, 0) + 1
print(freq)
print(sorted(freq.items()))
```

▶ Output

```
S a Lanka 9 SRI LANKA
[45, 62, 78, 90] [62, 78] 275
15
{'red': 3, 'blue': 1, 'green': 1}
[('blue', 1), ('green', 1), ('red', 3)]
```

🧱 **Worked example: Explain s[4:9], the sorted list and the word counts.**
"Sri Lanka" has 9 characters (index 0–8); s[4:9] takes 4 to 8 = **Lanka**. After append(90) and sort() the list is **[45, 62, 78, 90]**, the slice [1:3] is [62, 78] and the sum is **275**. The dictionary counts red **3**, blue 1, green 1.

- List methods: append, insert, remove, pop, sort, len().
- String methods: upper, lower, split, strip, find, replace.
- Dictionary: d[key], d.get(key, default), d.keys(), d.values(), d.items().

💡 Negative indexes count from the end: s[-1] is the last item.

🚀 **Beyond the syllabus (2026 update)** *Not required for the exam – read for interest.* JSON – the format most 2026 web APIs use – maps directly to Python dictionaries and lists.

Key Words

Data structure – Way of storing many values under one name.

String – Ordered immutable sequence of characters.

List – Ordered mutable sequence written in square brackets.

Tuple – Ordered immutable sequence written in round brackets.

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

Dictionary – Collection of key and value pairs.	**Index** – Position number of an item starting from 0.
Slice – Part of a sequence taken with start:stop.	**Mutable** – Can be changed after it is created.
Immutable – Cannot be changed after it is created.	**Method** – Function that belongs to an object such as append.
Key – Unique label used to look up a value in a dictionary.	**Concatenation** – Joining strings or lists with +.

Sample Q&A

1. List vs tuple?

> **Answer:** Lists can change; tuples cannot.

2. s[0] and s[-1] for "Sri Lanka"?

> **Answer:** S and a.

3. List after append(90) and sort()?

> **Answer:** [45, 62, 78, 90].

4. How many times does red appear?

> **Answer:** 3.

5. What does freq.get(w, 0) do?

> **Answer:** Returns the count or 0 if w is not yet a key.

6. Why use a tuple for a point?

> **Answer:** The coordinates should not change.

MCQ

1. For s = "Sri Lanka", s[4:9] is

A. Sri
B. anka
C. Lank
D. Sri L
E. Lanka

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

2. len("Sri Lanka") is

A. 8
B. 2
C. 9
D. 7
E. 10

3. After append(90) and sort(), marks[1:3] is

A. [45, 62, 78]
B. [62, 78]
C. [62, 78, 90]
D. [78, 90]
E. [45, 62]

4. sum of [45, 62, 78, 90] is

A. 230
B. 90
C. 275
D. 285
E. 265

5. Which structure cannot be changed after creation?

A. tuple
B. list
C. none
D. dictionary
E. set of lists

6. In the word count, freq["red"] is

A. 2
B. 5
C. 3
D. 0
E. 1

7. Key → value pairs are stored in a

A. file
B. dictionary
C. tuple
D. string
E. list only

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

8. point[0] + point[1] for (6, 9) is

A. 3
B. 6
C. 54
D. 69
E. 15

9. Which term means: “Way of storing many values under one name”?

A. Data structure
B. Mutable
C. Index
D. List
E. Key

10. Which term means: “Ordered immutable sequence of characters”?

A. String
B. Slice
C. Tuple
D. Immutable
E. Concatenation

11. Which term means: “Ordered mutable sequence written in square brackets”?

A. Method
B. Mutable
C. Data structure
D. List
E. Dictionary

12. Which term means: “Ordered immutable sequence written in round brackets”?

A. Index
B. Key
C. Tuple
D. String
E. Immutable

13. Which term means: “Collection of key and value pairs”?

A. Method
B. Dictionary
C. Concatenation
D. Slice
E. List

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

14. Which term means: “Position number of an item starting from 0”?

A. Mutable
B. Index
C. Key
D. Tuple
E. Data structure

15. Which term means: “Part of a sequence taken with start:stop”?

A. Dictionary
B. String
C. Concatenation
D. Slice
E. Immutable

16. Which term means: “Can be changed after it is created”?

A. Mutable
B. Data structure
C. List
D. Method
E. Index

17. Which term means: “Cannot be changed after it is created”?

A. Key
B. Tuple
C. String
D. Slice
E. Immutable

18. Which term means: “Function that belongs to an object such as append”?

A. List
B. Mutable
C. Concatenation
D. Method
E. Dictionary

19. Which term means: “Unique label used to look up a value in a dictionary”?

A. Tuple
B. Index
C. Data structure
D. Immutable
E. Key

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

20. Which term means: “Joining strings or lists with +”?

A. Method
B. Slice
C. Dictionary
D. Concatenation
E. String

Structured Questions

1. Compare strings, lists, tuples and dictionaries. *(4 marks)*

> Brackets; ordered; mutable or not; key → value.

2. Give the output of s[0], s[-1], s[4:9] and s.upper() for "Sri Lanka". *(4 marks)*

> S; a; Lanka; SRI LANKA.

3. Write code to add a mark, sort the list and print the total. *(3 marks)*

> marks.append(90); marks.sort(); print(sum(marks)).

4. Write code that counts words using a dictionary. *(4 marks)*

> split(); for each word; freq[w] = freq.get(w, 0) + 1; print.

5. Explain mutable and immutable with examples. *(2 marks)*

> List mutable (append); string/tuple immutable.

Summary

- Strings hold text and cannot be changed.
- Lists are ordered and changeable.
- Tuples are ordered and fixed.
- Dictionaries map keys to values.
- Indexes start at 0; slices exclude the stop.

MCQ answer key – 9.10

1. **E** – Index 4 to 8.
2. **C** – Space counts.
3. **B** – Items 1 and 2.
4. **C** – Total.
5. **A** – Tuple.
6. **C** – Three times.
7. **B** – Dictionary.

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

8. **E** – 6 + 9.
9. **A** – Data structure – Way of storing many values under one name.
10. **A** – String – Ordered immutable sequence of characters.
11. **D** – List – Ordered mutable sequence written in square brackets.
12. **C** – Tuple – Ordered immutable sequence written in round brackets.
13. **B** – Dictionary – Collection of key and value pairs.
14. **B** – Index – Position number of an item starting from 0.
15. **D** – Slice – Part of a sequence taken with start:stop.
16. **A** – Mutable – Can be changed after it is created.
17. **E** – Immutable – Cannot be changed after it is created.
18. **D** – Method – Function that belongs to an object such as append.
19. **E** – Key – Unique label used to look up a value in a dictionary.
20. **D** – Concatenation – Joining strings or lists with +.

© 2026 schoolict.net (Ruwan Mangala Suraweera). All rights reserved.

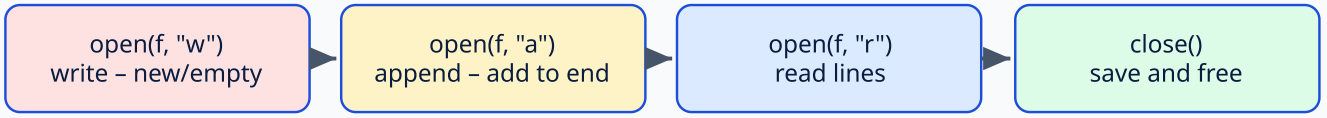
Grade 13 A/L ICT · Lesson 9

9.11 File Handling: Open, Read, Write, Append and Close

Syllabus 9.11. Variables are lost when a program ends; **files** keep data on storage. In Python you **open** a file with a **mode**, **read** or **write** it, and **close** it. Mode "w" creates or empties a file, "a" **appends** to the end, and "r" reads.

1. File modes

File modes



Opening in "w" mode erases an existing file – use "a" to keep old data.

The r, w and a modes

Mode	Meaning	If file is missing
"r"	Read from the start	Error
"w"	Write – old content is erased	Created
"a"	Append – add to the end	Created

🐍 Write, append and read a text file

```
f = open("marks.txt", "w")      # write: creates or empties
f.write("Nimal,78\n")
f.write("Kamala,45\n")
f.close()

f = open("marks.txt", "a")      # append: adds to the end
f.write("Sunil,88\n")
f.close()

f = open("marks.txt", "r")      # read
total = 0
for line in f:
    name, mark = line.strip().split(",")
    total += int(mark)
    print(name, mark)
f.close()
print("Total:", total)
```

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

▶ Output

```
Nimal 78
Kamala 45
Sunil 88
Total: 211
```

🧱 **Worked example: Why does the file have three lines and Total 211?**
"w" writes Nimal and Kamala; "a" keeps them and adds Sunil. Reading line by line, strip() removes the newline, split(",") separates name and mark, and int() converts it: 78 + 45 + 88 = **211**.

- read() returns the whole file; readline() one line; readlines() a list of lines.
- write() writes a string – add "\n" yourself for a new line.
- close() saves the buffer and frees the file.

💡 with open("marks.txt") as f: closes the file automatically, even if an error happens.

🚀 **Beyond the syllabus (2026 update)** *Not required for the exam – read for interest.* In 2026 many apps store data as CSV or JSON files; Python's csv and json modules read them in one line.

Key Words

File – Named collection of data kept on storage.	**open()** – Built-in function that opens a file in a mode.
close() – Method that finishes using a file and saves it.	**Read mode** – Mode r that reads an existing file.
Write mode – Mode w that creates or empties a file.	**Append mode** – Mode a that adds data to the end of a file.
read() – Method that returns the whole file as a string.	**readline()** – Method that returns one line of a file.
write() – Method that writes a string to a file.	**Newline** – Character \n that ends a line of text.
Text file – File of readable characters such as .txt or .csv.	**Persistent storage** – Storage that keeps data after the program ends.

Sample Q&A

1. Why use files?

Answer: Variables are lost when the program ends; files keep data.

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

2. "w" vs "a"?

> **Answer:** "w" erases old content; "a" adds to the end.

3. Opening a missing file with "r"?

> **Answer:** An error (FileNotFoundError).

4. Why strip() each line?

> **Answer:** To remove the newline before splitting.

5. Total printed?

> **Answer:** 211.

6. Why close a file?

> **Answer:** To save buffered data and free the file.

MCQ

1. Which mode adds data to the end of a file?

A. "a"
B. "x" only
C. "w"
D. "rb"
E. "r"

2. Opening an existing file with "w"

A. gives an error
B. locks it forever
C. adds to the end
D. erases its old content
E. only reads it

3. The Total printed by the files program is

A. 88
B. 211
C. 123
D. 200
E. 166

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

4. How many lines does marks.txt hold at the end?

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

5. line.strip().split(",") on "Nimal,78\n" gives

A. ("Nimal", 78)
B. ["Nimal", "78"]
C. "Nimal78"
D. ["Nimal,78"]
E. [78]

6. Which method returns one line?

A. open()
B. close()
C. read()
D. write()
E. readline()

7. Opening a missing file in "r" mode

A. creates it
B. raises an error
C. returns 0
D. appends
E. empties it

8. The with statement for files

A. closes the file automatically
B. sorts lines
C. opens two files
D. deletes the file
E. reads only binary

9. Which term means: “Named collection of data kept on storage”?

A. Text file
B. close()
C. File
D. readline()
E. Append mode

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

10. Which term means: “Built-in function that opens a file in a mode”?

A. open()
B. read()
C. Read mode
D. Persistent storage
E. write()

11. Which term means: “Method that finishes using a file and saves it”?

A. File
B. Write mode
C. Newline
D. readline()
E. close()

12. Which term means: “Mode r that reads an existing file”?

A. Append mode
B. open()
C. write()
D. Read mode
E. Text file

13. Which term means: “Mode w that creates or empties a file”?

A. Newline
B. close()
C. read()
D. Persistent storage
E. Write mode

14. Which term means: “Mode a that adds data to the end of a file”?

A. readline()
B. File
C. Read mode
D. Append mode
E. Text file

15. Which term means: “Method that returns the whole file as a string”?

A. Persistent storage
B. write()
C. open()
D. read()
E. Write mode

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

16. Which term means: “Method that returns one line of a file”?

A. readline()
B. Newline
C. close()
D. Append mode
E. File

17. Which term means: “Method that writes a string to a file”?

A. Read mode
B. write()
C. Text file
D. read()
E. open()

18. Which term means: “Character \n that ends a line of text”?

A. Persistent storage
B. close()
C. Write mode
D. readline()
E. Newline

19. Which term means: “File of readable characters such as .txt or .csv”?

A. File
B. write()
C. Text file
D. Read mode
E. Append mode

20. Which term means: “Storage that keeps data after the program ends”?

A. Write mode
B. read()
C. Persistent storage
D. Newline
E. open()

Structured Questions

1. Explain the r, w and a modes. *(3 marks)*

> r read; w create/erase; a append.

2. Write code to save three names to a file, one per line. *(3 marks)*

> open w; write each with \n; close.

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

3. Write code to read marks.txt and print the total. *(6 marks)*

> open r; loop lines; split; int; add; print.

4. Explain why the program prints Total 211. *(3 marks)*

> w two lines; a adds Sunil; 78 + 45 + 88.

5. State two reasons to close a file. *(2 marks)*

> Flush data; free the file for others.

Summary

- Files keep data after the program ends.
- open() with r, w or a.
- "w" erases; "a" appends.
- Read with read(), readline() or a for loop.
- Always close() or use with.

MCQ answer key – 9.11

1. **A** – Append.
2. **D** – Write empties.
3. **B** – 78 + 45 + 88.
4. **C** – Two written, one appended.
5. **B** – Two strings.
6. **E** – readline.
7. **B** – FileNotFoundError.
8. **A** – Auto close.
9. **C** – File – Named collection of data kept on storage.
10. **A** – open() – Built-in function that opens a file in a mode.
11. **E** – close() – Method that finishes using a file and saves it.
12. **D** – Read mode – Mode r that reads an existing file.
13. **E** – Write mode – Mode w that creates or empties a file.
14. **D** – Append mode – Mode a that adds data to the end of a file.
15. **D** – read() – Method that returns the whole file as a string.
16. **A** – readline() – Method that returns one line of a file.
17. **B** – write() – Method that writes a string to a file.
18. **E** – Newline – Character \n that ends a line of text.
19. **C** – Text file – File of readable characters such as .txt or .csv.
20. **C** – Persistent storage – Storage that keeps data after the program ends.

© 2026 schoolict.net (Ruwan Mangala Suraweera). All rights reserved.

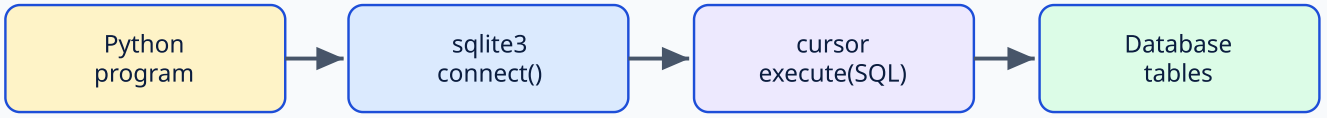
Grade 13 A/L ICT · Lesson 9

9.12 Python and Databases: Connect, Add, Modify, Delete, Retrieve

Syllabus 9.12. A program can **connect** to a database and run SQL to **add**, **modify**, **delete** and **retrieve** records. Python's built-in **sqlite3** module needs no server: connect() opens the database, a **cursor** runs SQL with execute(), commit() saves changes and close() disconnects.

1. Steps

Python program and a database



INSERT, UPDATE, DELETE need commit(); SELECT rows come back as tuples.

Python program talking to a database

- **Connect**: con = sqlite3.connect("school.db").
- **Cursor**: cur = con.cursor() runs SQL statements.
- **Add / modify / delete**: INSERT, UPDATE, DELETE then con.commit().
- **Retrieve**: SELECT and loop over the rows; then con.close().

🐍 Python with an SQLite database

```
import sqlite3
con = sqlite3.connect(":memory:")       # connect
cur = con.cursor()
cur.execute("CREATE TABLE Student (ID TEXT PRIMARY KEY, Name TEXT, Mark INTEGER)")
rows = [("S001", "Nimal", 78), ("S002", "Kamala", 45), ("S003", "Sunil", 88)]
cur.executemany("INSERT INTO Student VALUES (?, ?, ?)", rows)       # add
cur.execute("UPDATE Student SET Mark = 50 WHERE ID = ?", ("S002",))  # modify
cur.execute("DELETE FROM Student WHERE ID = ?", ("S003",))           # delete
con.commit()
for row in cur.execute("SELECT Name, Mark FROM Student ORDER BY Name"):  # retrieve
    print(row)
con.close()
```

▶ Output

```
('Kamala', 50)
('Nimal', 78)
```

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

🧮 **Worked example: Why are only Kamala 50 and Nimal 78 printed?**
Three rows are inserted. UPDATE changes S002's mark from 45 to **50**; DELETE removes S003 (Sunil). SELECT … ORDER BY Name returns the two rows left in name order.

💡 Use ? placeholders, never string joining, to put values into SQL – this stops SQL injection.

🚀 **Beyond the syllabus (2026 update)** *Not required for the exam – read for interest.* Web apps in 2026 often use an ORM such as SQLAlchemy or Django, which writes this SQL for you, but the same connect–execute–commit idea lies underneath.

Key Words

Database connection – Link between a program and a database.	**sqlite3** – Built-in Python module for SQLite databases.
connect() – Function that opens a database connection.	**Cursor** – Object that executes SQL and holds results.
execute() – Method that runs one SQL statement.	**commit()** – Method that saves changes permanently.
INSERT – SQL command that adds a record.	**UPDATE** – SQL command that modifies records.
DELETE – SQL command that removes records.	**SELECT** – SQL command that retrieves records.
Placeholder – The ? mark that safely takes a value in SQL.	**SQL injection** – Attack that inserts harmful SQL through input.

Sample Q&A

1. Four operations from Python?

> **Answer:** Add, modify, delete, retrieve.

2. Role of a cursor?

> **Answer:** It executes SQL and returns rows.

3. Why commit()?

> **Answer:** Changes are not saved until commit.

4. Kamala's mark after UPDATE?

> **Answer:** 50.

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

5. How many rows does SELECT return?

> **Answer:** 2.

6. Why use ? placeholders?

> **Answer:** Safety against SQL injection.

MCQ

1. Python's built-in module for SQLite is

A. sqlite3
B. mysql
C. os
D. csv
E. pandas

2. Which object runs SQL statements?

A. connection string
B. cursor
C. print
D. table
E. commit

3. Changes are saved permanently by

A. commit()
B. cursor()
C. execute()
D. fetch()
E. close() only

4. After the program, Kamala's mark is

A. 88
B. 45
C. 0
D. 50
E. 78

5. Which record is removed by DELETE?

A. S003 Sunil
B. none
C. S002 Kamala
D. all records
E. S001 Nimal

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

6. The first row printed is

A. ('S001', 'Nimal')
B. ('Sunil', 88)
C. ('Kamala', 45)
D. ('Nimal', 78)
E. ('Kamala', 50)

7. Retrieving records uses

A. UPDATE
B. INSERT
C. DELETE
D. SELECT
E. DROP

8. ? placeholders mainly protect against

A. overflow
B. syntax colouring
C. slow disks
D. viruses in files
E. SQL injection

9. Which term means: “Link between a program and a database”?

A. UPDATE
B. commit()
C. Database connection
D. Placeholder
E. connect()

10. Which term means: “Built-in Python module for SQLite databases”?

A. Cursor
B. sqlite3
C. SQL injection
D. INSERT
E. DELETE

11. Which term means: “Function that opens a database connection”?

A. SELECT
B. execute()
C. connect()
D. UPDATE
E. Database connection

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

12. Which term means: “Object that executes SQL and holds results”?

A. commit()
B. Placeholder
C. DELETE
D. sqlite3
E. Cursor

13. Which term means: “Method that runs one SQL statement”?

A. SELECT
B. execute()
C. INSERT
D. SQL injection
E. connect()

14. Which term means: “Method that saves changes permanently”?

A. Database connection
B. commit()
C. UPDATE
D. Placeholder
E. Cursor

15. Which term means: “SQL command that adds a record”?

A. sqlite3
B. execute()
C. DELETE
D. INSERT
E. SQL injection

16. Which term means: “SQL command that modifies records”?

A. UPDATE
B. commit()
C. SELECT
D. Database connection
E. connect()

17. Which term means: “SQL command that removes records”?

A. sqlite3
B. INSERT
C. DELETE
D. Cursor
E. Placeholder

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

18. Which term means: “SQL command that retrieves records”?

A. SQL injection
B. UPDATE
C. connect()
D. execute()
E. SELECT

19. Which term means: “The ? mark that safely takes a value in SQL”?

A. commit()
B. Database connection
C. Placeholder
D. Cursor
E. DELETE

20. Which term means: “Attack that inserts harmful SQL through input”?

A. SELECT
B. execute()
C. sqlite3
D. SQL injection
E. INSERT

Structured Questions

1. List the steps to use a database from Python. *(5 marks)*

> connect; cursor; execute; commit; close.

2. Write code to insert a student and display all students. *(4 marks)*

> execute INSERT with ?; commit; for row in SELECT; print.

3. Explain why the program prints two rows only. *(4 marks)*

> Three inserted; one deleted; one updated; order.

4. Explain the purpose of commit() and close(). *(2 marks)*

> commit saves; close frees the connection.

5. What is SQL injection and how is it prevented? *(2 marks)*

> Harmful input in SQL; use placeholders.

Summary

- connect() opens the database.

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

- A cursor executes SQL.
- INSERT, UPDATE, DELETE change data.
- SELECT retrieves rows.
- commit() saves; use ? for safety.

MCQ answer key – 9.12

1. **A** – sqlite3.
2. **B** – Cursor.
3. **A** – Commit.
4. **D** – UPDATE.
5. **A** – WHERE ID = S003.
6. **E** – ORDER BY Name.
7. **D** – SELECT.
8. **E** – Injection.
9. **C** – Database connection – Link between a program and a database.
10. **B** – sqlite3 – Built-in Python module for SQLite databases.
11. **C** – connect() – Function that opens a database connection.
12. **E** – Cursor – Object that executes SQL and holds results.
13. **B** – execute() – Method that runs one SQL statement.
14. **B** – commit() – Method that saves changes permanently.
15. **D** – INSERT – SQL command that adds a record.
16. **A** – UPDATE – SQL command that modifies records.
17. **C** – DELETE – SQL command that removes records.
18. **E** – SELECT – SQL command that retrieves records.
19. **C** – Placeholder – The ? mark that safely takes a value in SQL.
20. **D** – SQL injection – Attack that inserts harmful SQL through input.

© 2026 schoolict.net (Ruwan Mangala Suraweera). All rights reserved.

Grade 13 A/L ICT · Lesson 9

9.13 Searching and Sorting: Sequential Search and Bubble Sort

Syllabus 9.13. Sequential (linear) search checks items one by one from the start until the target is found or the list ends. **Bubble sort** compares neighbouring items and swaps them if they are in the wrong order; after each **pass** the largest remaining value “bubbles” to the end.

1. Sequential search

🐍 Sequential search

```
def linear_search(items, target):
    for i in range(len(items)):
        if items[i] == target:
            return i
    return -1

data = [34, 7, 23, 32, 5, 62]
print(linear_search(data, 23))
print(linear_search(data, 99))
```

▶ Output

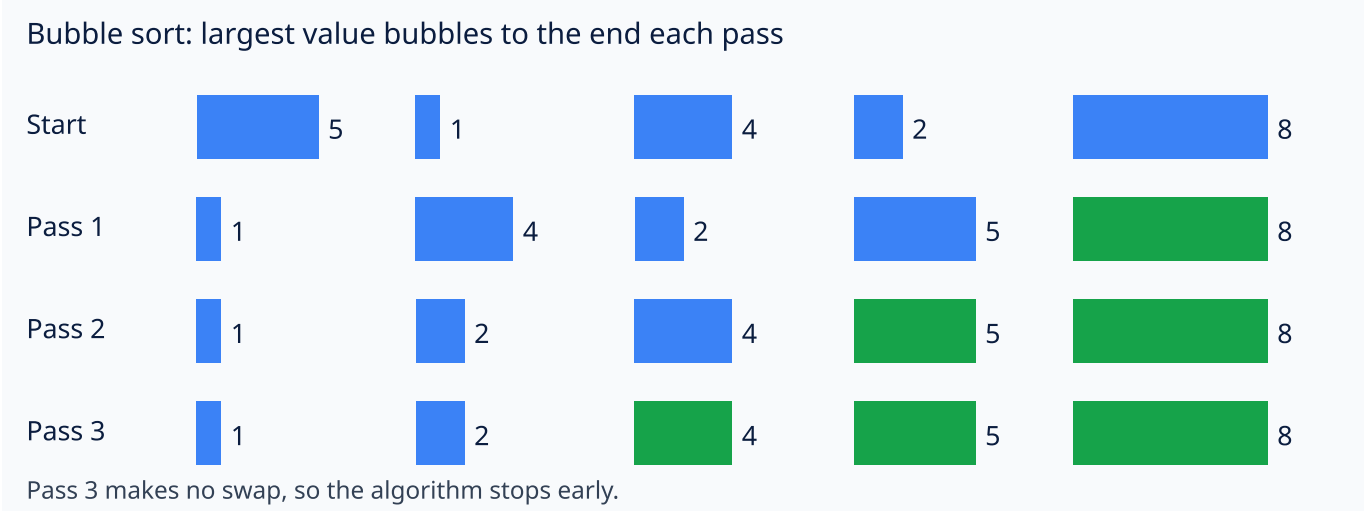
```
2
-1
```

Target	Result index	Comparisons
23	2	3
99	−1	6

🧱 **Worked example: How many comparisons to find 23 and 99 in [34, 7, 23, 32, 5, 62]?**
23 is checked against 34, 7, 23 – found at index **2** after **3** comparisons. 99 is compared with all **6** items and the function returns **−1**.

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

2. Bubble sort



Pass 1 of bubble sort

🐍 Bubble sort showing each pass

```
def bubble_sort(a):
    n = len(a)
    for p in range(n - 1):
        swapped = False
        for j in range(n - 1 - p):
            if a[j] > a[j + 1]:
                a[j], a[j + 1] = a[j + 1], a[j]
                swapped = True
        print("Pass", p + 1, a)
        if not swapped:
            break
    return a

bubble_sort([5, 1, 4, 2, 8])
```

▶ Output

```
Pass 1 [1, 4, 2, 5, 8]
Pass 2 [1, 2, 4, 5, 8]
Pass 3 [1, 2, 4, 5, 8]
```

Pass	List after the pass
1	[1, 4, 2, 5, 8]
2	[1, 2, 4, 5, 8]
3	[1, 2, 4, 5, 8]

🧮 Worked example: Sort [5, 1, 4, 2, 8] with bubble sort.
Pass 1 swaps 5↔1, 5↔4, 5↔2 → [1, 4, 2, 5, 8]. Pass 2 swaps 4↔2 → [1, 2, 4, 5, 8]. Pass 3 makes no swap, so the swapped flag stops the sort early. Total swaps: **4**.

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

💡 For n items bubble sort needs at most n − 1 passes; sequential search needs at most n comparisons.

🚀 **Beyond the syllabus (2026 update)** *Not required for the exam – read for interest.* Python's built-in sorted() uses Timsort, much faster than bubble sort on large data – but bubble sort is the classic for learning how sorting works.

Key Words

Searching – Finding whether and where a value is in a list.	**Sequential search** – Checking each item in turn from the start.
Target – Value being searched for.	**Comparison** – Checking one item against the target or a neighbour.
Sorting – Arranging items in ascending or descending order.	**Bubble sort** – Sort that swaps neighbouring items in the wrong order.
Pass – One full run through the list in bubble sort.	**Swap** – Exchanging the positions of two items.
Swapped flag – Boolean that stops the sort when a pass has no swaps.	**Ascending order** – Smallest to largest.
Return −1 – Signal that the target was not found.	**Algorithm efficiency** – How the work grows as the data grows.

Sample Q&A

1. How does sequential search work?

> **Answer:** Checks items one by one until found or the end.

2. Index of 23?

> **Answer:** 2.

3. Comparisons for 99?

> **Answer:** 6 (all items), returns −1.

4. List after pass 1?

> **Answer:** [1, 4, 2, 5, 8].

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

5. Why does it stop after pass 3?

> **Answer:** No swaps, so swapped stays False.

6. Maximum passes for 5 items?

> **Answer:** 4 (n − 1).

MCQ

1. Sequential search for 23 in [34, 7, 23, 32, 5, 62] returns

A. 2
B. −1
C. 3
D. 23
E. 1

2. Comparisons needed to find 23 are

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

3. Searching for 99 returns

A. 6
B. 99
C. −1
D. None
E. 0

4. After pass 1, [5, 1, 4, 2, 8] becomes

A. [5, 4, 2, 1, 8]
B. [1, 4, 2, 5, 8]
C. [8, 5, 4, 2, 1]
D. [1, 2, 4, 5, 8]
E. [1, 5, 4, 2, 8]

5. Total swaps sorting [5, 1, 4, 2, 8] are

A. 4
B. 3
C. 5
D. 6
E. 2

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

6. Bubble sort compares

A. random pairs
B. every item with the target
C. the middle item
D. first and last only
E. neighbouring items

7. The swapped flag lets bubble sort

A. swap faster
B. skip pass 1
C. search for a value
D. stop early when a pass has no swaps
E. sort descending

8. The maximum passes for n items is

A. n + 1
B. n^2
C. n − 1
D. 2n
E. n

9. Which term means: “Finding whether and where a value is in a list”?

A. Bubble sort
B. Return −1
C. Swap
D. Target
E. Searching

10. Which term means: “Checking each item in turn from the start”?

A. Pass
B. Sequential search
C. Swapped flag
D. Comparison
E. Algorithm efficiency

11. Which term means: “Value being searched for”?

A. Target
B. Searching
C. Ascending order
D. Swap
E. Sorting

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

12. Which term means: “Checking one item against the target or a neighbour”?

A. Swapped flag
B. Return −1
C. Comparison
D. Bubble sort
E. Sequential search

13. Which term means: “Arranging items in ascending or descending order”?

A. Ascending order
B. Sorting
C. Target
D. Algorithm efficiency
E. Pass

14. Which term means: “Sort that swaps neighbouring items in the wrong order”?

A. Return −1
B. Bubble sort
C. Comparison
D. Swap
E. Searching

15. Which term means: “One full run through the list in bubble sort”?

A. Algorithm efficiency
B. Sequential search
C. Pass
D. Swapped flag
E. Sorting

16. Which term means: “Exchanging the positions of two items”?

A. Swap
B. Bubble sort
C. Searching
D. Target
E. Ascending order

17. Which term means: “Boolean that stops the sort when a pass has no swaps”?

A. Sequential search
B. Return −1
C. Pass
D. Comparison
E. Swapped flag

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

18. Which term means: “Smallest to largest”?

A. Algorithm efficiency
B. Sorting
C. Target
D. Ascending order
E. Swap

19. Which term means: “Signal that the target was not found”?

A. Comparison
B. Swapped flag
C. Searching
D. Bubble sort
E. Return −1

20. Which term means: “How the work grows as the data grows”?

A. Sequential search
B. Ascending order
C. Sorting
D. Algorithm efficiency
E. Pass

Structured Questions

1. Write a Python function for sequential search. *(3 marks)*

> for i in range(len); if equal return i; return −1.

2. Find 23 and 99 in [34, 7, 23, 32, 5, 62] with comparison counts. *(2 marks)*

> 23: index 2, 3 comparisons; 99: −1, 6 comparisons.

3. Show each pass of bubble sort on [5, 1, 4, 2, 8]. *(3 marks)*

> [1, 4, 2, 5, 8]; [1, 2, 4, 5, 8]; no swaps, stop.

4. Explain the purpose of the swapped flag. *(3 marks)*

> Detects a pass with no swaps; ends early; saves time.

5. Compare the best and worst case of sequential search. *(2 marks)*

> Best 1 comparison (first item); worst n (last or missing).

Summary

- Sequential search checks items in order.

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

- It returns the index or −1.
- Bubble sort swaps neighbours in the wrong order.
- Each pass puts the largest left value at the end.
- A pass with no swaps ends the sort.

MCQ answer key – 9.13

1. **A** – Index from 0.
2. **D** – 34, 7, 23.
3. **C** – Not found.
4. **B** – 5 bubbles to its place.
5. **A** – 3 + 1.
6. **E** – Adjacent.
7. **D** – Early exit.
8. **C** – n − 1.
9. **E** – Searching – Finding whether and where a value is in a list.
10. **B** – Sequential search – Checking each item in turn from the start.
11. **A** – Target – Value being searched for.
12. **C** – Comparison – Checking one item against the target or a neighbour.
13. **B** – Sorting – Arranging items in ascending or descending order.
14. **B** – Bubble sort – Sort that swaps neighbouring items in the wrong order.
15. **C** – Pass – One full run through the list in bubble sort.
16. **A** – Swap – Exchanging the positions of two items.
17. **E** – Swapped flag – Boolean that stops the sort when a pass has no swaps.
18. **D** – Ascending order – Smallest to largest.
19. **E** – Return −1 – Signal that the target was not found.
20. **D** – Algorithm efficiency – How the work grows as the data grows.

© 2026 schoolict.net (Ruwan Mangala Suraweera). All rights reserved.