Computer Science

Exam — **ANSWER KEY**

Q1. L1 Which of the following best describes **Decomposition**? **[1 pt]**

B — Breaking a complex problem into smaller, more manageable sub-problems.

Marking note: A describes Abstraction; C describes an Algorithm; D describes Programming.

Q2. L2 In a flowchart, which shape is used to represent a **decision**? **[1 pt]**

C — Diamond.

Marking note: Rectangle = process; Oval = start/end; Parallelogram = input/output.

Q3. L3 What is the result of `9 % 4` in Python? **[1 pt]**

C — 1

Explanation: $9 = 4 \times 2 + 1$*, so the remainder is 1. The* `%` *operator returns the remainder of integer division.*

Q4. L3 Which of the following is a **valid** Python variable name? **[1 pt]**

C — `_result`

Marking note: `3value` *starts with a digit (invalid);* `my value` *contains a space (invalid);* `while` *is a reserved keyword (invalid).*

Q5. **[4 pts]**

a. L1 Abstraction means keeping *all* details of a problem, including unnecessary ones.

FALSE — Abstraction is the process of *removing* unnecessary details and focusing only on what is relevant to solve the problem.

b. L2 A WHILE loop executes its body at least once before checking the condition.

FALSE — A WHILE loop checks the condition *before* each iteration. If the condition is false from the start, the body never executes. (It is the DO-WHILE / REPEAT-UNTIL loop that runs at least once.)

c. L2 The three structured programming control structures are: sequence, selection, and repetition.

TRUE — These are indeed the three fundamental control structures: sequence (instructions executed in order), selection (IF/ELSE branching), and repetition (FOR/WHILE loops).

d. L3 In Python, `Name` and `name` refer to the same variable.

FALSE — Python is case-sensitive. `Name` and `name` are two completely different variables.

Q6. L2 **[4 pts]**

```
BEGIN
  PRINT "Enter N:"
  READ N
  sum     = 0
  counter = 1
  WHILE counter <= N
      sum     = sum + counter
      counter = counter + 1
  END_WHILE
  PRINT sum
END
```

a. What does this algorithm compute?

This algorithm computes the sum of all integers from 1 to N.
i.e., sum $= 1 + 2 + 3 + \cdots + N = \dfrac{N(N+1)}{2}$

b. Trace table for N = 4:

N	sum	counter	counter $\leq$ N
4	0	1	T
4	**1**	**2**	**T**
4	**3**	**3**	**T**
4	**6**	**4**	**T**
4	**10**	**5**	**F**
Loop exits because counter (5) > N (4)			

c. What value is printed at the end?

10 $(1 + 2 + 3 + 4 = 10)$

Q7. L1 L2 Pseudocode for finding the largest of two numbers + explanation. **[4 pts]**

Pseudocode algorithm:

```
BEGIN
  PRINT "Enter first number:"
  READ A
  PRINT "Enter second number:"
  READ B
  IF A > B THEN
      PRINT A
  ELSE
      PRINT B
  END_IF
END
```

Abstraction: We ignore unimportant details (e.g., the type of numbers, memory management) and focus only on the comparison logic needed to find the maximum.

Algorithmic Thinking: We break the problem into clear, ordered steps: read inputs → compare → output result.

Q8. L3 Output and explanation of the program. **[2 pts]**

```
a = 14
b = 4
print("Floor div:", a // b)
print("Remainder:", a % b)
print("Power:",      b ** 3)
text = "Algorithm"
print(len(text))
print(text[0])
```

Output	Explanation
`Floor div:  3`	$14 \div 4 = 3$ remainder 2; `//` gives the integer quotient
`Remainder:  2`	$14 \mod 4 = 2$; `%` gives the remainder
`Power:  64`	$4^3 = 64$; `**` is the exponentiation operator
`9`	`"Algorithm"` has 9 characters; `len()` counts them
`A`	Index 0 is the first character of the string `"Algorithm"`

Q9. L3 Find and fix the 3 errors. **[2 pts]**

```
my grade = int(input("Enter grade: "))
If my_grade >= 50
    print("Pass")
```

#	Error	Correction
1	`my grade` contains a space — invalid variable name	Rename to `my_grade`
2	`If` — Python keywords are case-sensitive	Change to `if` (lowercase)
3	Missing colon at the end of the `if` statement	Add `:` after the condition

Corrected program:

```
my_grade = int(input("Enter grade: "))
if my_grade >= 50:
    print("Pass")
```