

ANSWER KEY & MARKING SCHEME · CBSE CLASS 12**Python Revision Tour – I**

Computer Science · Chapter 1 · Use this with the Board Paper · Companion to Quick Drill

HOW TO USE

Attempt the Board Paper first (closed-book, full time). Then come here. For 2-mark+ questions, compare your answer to the model. For 3-4 mark questions, also consult the **Topper Templates** below — these show the exact step-by-step structure that scores full marks per CBSE marking-scheme conventions.

MODEL ANSWERS · BOARD PAPER**Section A — Short (2 × 4 = 8 marks)****Q1. Predict the output: `print(15 // 4, 15 % 4, 15 / 4, 2 ** 3)` [2 marks]**

| Ans: 3 3 3.75 8

Q2. Differentiate between mutable and immutable types in Python with one example each. [2 marks]| Ans: Mutable types (list, dict, set) can be modified in place. Immutable types (int, str, tuple) cannot — operations create new objects. Example: a list `.append()` modifies in place; a string `+` operation creates a new string.**Q3. Identify and rewrite the error: `if x = 5: print('five')` [2 marks]**| Ans: Error: `=` (assignment) instead of `==` (comparison) inside if-condition. Corrected: `if x == 5: print('five')`**Q4. Predict the output: `s = 'PYTHON'; print(s[1:4], s[::-1])` [2 marks]**

| Ans: YTH NOHTYP

Section B — Output prediction (3 × 4 = 12 marks)**Q5. Predict the output: `for i in range(1, 6): if i == 3: continue print(i, end=' ')` [3 marks]**

| Ans: 1 2 4 5

Q6. Predict the output: `a = [1, 2, 3]; b = a; b.append(4); print(a)` [3 marks]

| Ans: [1, 2, 3, 4] — because a and b refer to the same list object (mutable).

Q7. Predict the output: `x = 5; y = x; x = x + 10; print(x, y)` [3 marks]

| Ans: 15 5 — int is immutable; x rebinds to new object 15; y still references original 5.

Q8. Predict the output: `for i in range(2, 11, 2): print(i * i, end=' ')` [3 marks]

| Ans: 4 16 36 64 100

Section C — Programs (5 × 2 = 10 marks)**Q9. Write a Python program to read N numbers from the user and print their sum and average. [5 marks]**| Ans: `N = int(input('How many numbers? ')) total = 0 for i in range(N): x = float(input(f'Number {i+1}: ')) total += x print('Sum:', total, 'Average:', total / N)`**Q10. Write a Python program that reads a string and counts the number of vowels and consonants. Print both counts. [5 marks]**| Ans: `s = input('Enter a string: ').lower() vowels = consonants = 0 for ch in s: if ch in 'aeiou': vowels += 1 elif ch.isalpha(): consonants += 1 print('Vowels:', vowels, 'Consonants:', consonants)`

★ **TOPPER TEMPLATE — 2 marks: Predict the output of the following Python code: `x = 5; y = 2; print(x // y, x / y, x ** y)`**

Annual

Step 1 [0.5 mark]	Identify each operator	Three operators in play: // is FLOOR DIVISION (integer-truncated quotient toward minus infinity), / is TRUE DIVISION (always returns float), ** is EXPONENTIATION (x to the power y).
Step 2 [1 mark]	Compute each value	$x // y = 5 // 2 = 2$ (integer truncation of 2.5). $x / y = 5 / 2 = 2.5$ (float result, even though both inputs were ints). $x ** y = 5 ** 2 = 25$.
Step 3 [0.5 mark]	Write the output exactly as Python prints it	print() with comma-separated arguments joins them with a single space. So the output is exactly: 2 2.5 25. Note that Python prints 2.5 not 2.50, and the values are space-separated, not comma-separated.

COMMON LOSS OF MARKS:

- Returning 2 (int) instead of 2.5 (float) for true division — the most common error.
- Writing the output with commas (the print() arguments — but print() outputs spaces between them).
- Forgetting that $5 ** 2 = 25$ and writing 10 (confusing ** with *).

★ **TOPPER TEMPLATE — 3 marks: Predict the output of: `for i in range(1, 6): if i == 3: continue; print(i, end='')`**

Annual

Step 1 [1 mark]	Translate range(1, 6) to its actual values	range(1, 6) generates the integers 1, 2, 3, 4, 5 (stop value 6 is excluded). So the loop body runs once for each of these five values, with the loop variable i taking those values in turn.
Step 2 [1 mark]	Apply the if i == 3: continue	The 'continue' statement skips the rest of the current loop iteration and jumps to the next. So when i is 3, the print line below is skipped. For i = 1, 2, 4, 5 the print runs normally.
Step 3 [1 mark]	Write the final output	Because print uses end=' ' (single space, not newline), the values print on one line separated by spaces. Final output: '1 2 4 5 ' (a trailing space follows the 5). Note that 3 is missing because of the continue.

COMMON LOSS OF MARKS:

- Including 3 in the output (forgetting that continue skips the print).
- Including 6 in the output (forgetting that range(1, 6) stops at 5).
- Putting each value on a new line (forgetting that end=' ' overrides the default newline).

★ **TOPPER TEMPLATE — 3 marks: Write a Python program to input N numbers from the user and print the largest.**

Annual

Step 1 [1 mark]	Read N	First read N from the user, converting the input to an integer: <code>N = int(input('How many numbers? '))</code> Without the int() cast, input() returns a string and the range() below would fail.
Step 2 [1.5 marks]	Read N numbers and track the largest	Initialise largest with the FIRST input, then compare against subsequent inputs: <code>largest = float(input('Number 1: '))</code> for i in range(2, N+1): <code>x = float(input(f'Number {i}: '))</code> if <code>x > largest: largest = x</code> This pattern avoids the trap of initialising largest to 0 (fails when all inputs are negative) or $-\infty$ (avoidable complication).
Step 3 [0.5 mark]	Print the answer	<code>print('Largest:', largest)</code> Using print's automatic space-join produces clean output. Always include a descriptive label — a bare number is not full-mark output.

COMMON LOSS OF MARKS:

- Forgetting the int() cast on input() — board examiners deduct half a mark for this.
- Initialising largest = 0 — fails when all inputs are negative.
- Using a hard-coded loop limit (range(10)) instead of using N as the bound.
- No descriptive output label — bare value loses presentation marks.

MARKING SCHEME — GENERAL NOTES

- For output-prediction: write the output exactly as Python prints it (spaces, no commas between args, no quotes around strings printed with print()).
- For programs: input() casts to int/float are MANDATORY for numerical input — half a mark deducted if missing.
- For programs: always print a descriptive label with the answer ('Sum:' not just the number).
- Indentation: use 4 spaces consistently throughout; mixing with tabs causes TabError in real code (and loses marks).
- For 'identify and rewrite' questions: state the error AND give the corrected line — half marks for each.