

## Operator Precedence in Python – Past G.C.E. A/L Questions and Model MCQs

### Past G.C.E. A/L MCQs on Operator Precedence

The following table summarises three multiple-choice questions from past G.C.E. A/L Information & Communication Technology examinations (new syllabus) that require students to evaluate Python expressions or trace code where operator precedence is important. The year and question number are given for reference.

Year & question no.	Question (extract)	Options (original paper)	Evidence
<b>2019 (new syllabus) – Q44</b>	<i>What is the value of the following Python expression?</i>		
(100 // 3) % 4 / 8	1) 0   2) 0.125   3) 3 4) 8   5) 9	The 2019 paper contains a question asking candidates to evaluate (100 // 3) % 4 / 8. The printed options show that (100 // 3) is computed first, then the modulo % operation, and finally the division /. The question appears on page 10 of the 2019 MCQ paper	【675507043435519† screenshot】 .
<b>2018 (new syllabus) – Q35</b>	<i>What is the value of the following Python expression?</i>		
(5**2) // 3 ^ 4	1) 12   2) 4   3) 8   4) 9 5) 16	The 2018 English-medium paper asks candidates to evaluate the expression (5**2) // 3 ^ 4. The extracted PDF text shows the question	

Year & question no.	Question (extract)	Options (original paper)	Evidence
		heading “What is the value of the following Python expression?” 【38059 8288119615†L340-L352】 . The expression uses exponentiation (**), floor-division (//) and the bitwise XOR operator (^), so correct evaluation requires knowing Python’s operator precedence (exponentiation has the highest precedence, followed by floor division and bitwise XOR).	
<b>2019 (new syllabus) – Q46</b>	<i>What will be the output of the following Python code?</i>		

```
x = 100
for i in range(1,5):
    x = x - i
print(x)
```

“ | 1) 0 2) 5 3) 85 4) 90 5) 100 | The 2019 paper also contains a question where students must trace a loop that subtracts successive integers **from** `100`. Although this question focuses on loop iteration, it relies on understanding that subtraction (`-`) **is** evaluated before assignment, which **is** part of Python’s arithmetic-operator precedence. The question **and** options are printed below question 45 on the same page of the 2019 MCQ paper 【675507043435519†screenshot】 . |

### *## Model MCQs on Python Operator Precedence (for 2025 A/L exam)*

The following ten questions are designed to **help** you practise Python’s operator-precedence rules. Each MCQ has five answer choices; only one choice **is** correct.

1. **\*\***Evaluate the expression below:**\*\***

```
python
7 + 3 * 2 ** 2
```

- 1) 13   2) 19   3) 16   4) 22   5) 25

**Answer:** 2) 19

*Explanation:*  $2 ** 2 = 4$ , then  $3 * 4 = 12$ , and finally  $7 + 12 = 19$ .

2. What is the result of the following expression?

```
(5 + 3) // 2 ** 2
```

- 1) 1   2) 2   3) 4   4) 3   5) 0

**Answer:** 2) 2

*Explanation:* parenthesis gives  $5+3=8$ ; exponentiation  $2**2=4$ ; floor-division  $8//4=2$ .

3. Consider the expression:

```
5 * 2 ** 3 % 7
```

- 1) 1   2) 2   3) 4   4) 5   5) 6

**Answer:** 3) 4

*Explanation:*  $2**3=8$ ;  $5*8=40$ ;  $40\%7=4$

4. Find the value of the expression:

```
4 + 18 // 3 ** 2
```

- 1) 4   2) 6   3) 10   4) 7   5) 16

**Answer:** 2) 6

*Explanation:*  $3**2=9$ ;  $18//9=2$ ;  $4+2=6$ .

5. What does the following expression evaluate to?

```
2 ** 3 ** 2
```

- 1) 64   2) 512   3) 16   4) 256   5) 8

**Answer:** 2) 512

*Explanation:* exponentiation is right-associative; compute  $3**2=9$  first, then  $2**9=512$ .

6. Evaluate:

```
16 >> 2 + 1
```

- 1) 1   2) 2   3) 4   4) 8   5) 16

**Answer:** 2) 2

*Explanation:* Addition has higher precedence than bit-shifting, so Python interprets the expression as  $16 \gg (2 + 1)$ . First compute  $2 + 1 = 3$ ; then perform the right-shift:  $16 \gg 3 = 2$ .

7. **What is the value of this expression?**

$3 + 4 * 2 ** 2 // 5$

1) 4   2) 5   3) 6   4) 7   5) 9

**Answer:** 3) 6

*Explanation:*  $2 ** 2 = 4$ ; then  $4 * 4 = 16$ ;  $16 // 5 = 3$ ; finally  $3 + 3 = 6$ . Thus the correct option is **3**.

8. **Determine the output:**

$9 \% 2 ** 3$

1) 1   2) 3   3) 5   4) 7   5) 0

**Answer:** 1) 1

*Explanation:* Exponentiation is evaluated first:  $2 ** 3 = 8$ . The modulus operator has lower precedence, so the expression becomes  $9 \% 8 = 1$ . Therefore the correct option is **1**.

9. **What will be printed by the following code?**

```
a = 2
b = 3
c = 4
print(a + b * c ** b % a)
```

1) 0   2) 2   3) 3   4) 5   5) 6

**Answer:** 2) 2

*Explanation:*  $c ** b = 4 ** 3 = 64$ ;  $b * 64 = 192$ ;  $192 \% a = 192 \% 2 = 0$ ; then  $a + 0 = 2$ .

10. **If x = 4, y = 2, and z = 3, what does the following expression return?**

$x ** y // z + x \% y$

1) 4   2) 5   3) 6   4) 7   5) 8

**Answer:** 2) 5

*Explanation:*  $x ** y = 4 ** 2 = 16$ ;  $16 // z = 16 // 3 = 5$ ;  $x \% y = 4 \% 2 = 0$ ; sum is  $5 + 0 = 5$ .

---

## Notes

- Operator precedence in Python follows a specific order: exponentiation ( $**$ ) is evaluated first, then unary operators, multiplication/division/floor division/modulo, addition/subtraction, bitwise shifts, bitwise AND/OR/XOR, comparisons, boolean operations, and finally assignment. Parentheses can be used to override the default order.
- In the 2019 MCQ question shown above,  $(100 // 3) \% 4 / 8$  first performs floor-division ( $//$ ), then modulo ( $\%$ ), and lastly floating-point division ( $/$ ), yielding **0.125** [【675507043435519†screenshot】](#) .
- In the 2018 MCQ, exponentiation has higher precedence than floor division, which in turn precedes the bitwise XOR operator. The expression  $(5**2) // 3 \wedge 4$  therefore evaluates as  $25 // 3 = 8$ , then  $8 \wedge 4 = 12$  [【380598288119615†L340-L352】](#) .