

Q4

Performance Report

PYTHON MATCH STATEMENT EXAM

✅ PYTHON MATCH STATEMENT EXAM – RESULTS

🎯 SECTION 1: MULTIPLE CHOICE QUESTIONS (5 points total)

Q1. What does the match statement do?

✅ **Your Answer:** B – Correct

📄 **Explanation:** The match statement compares an expression to multiple case patterns. Only one evaluation — very efficient.

Score: 1/1

Q2. What version introduced match?

✅ **Your Answer:** C (Python 3.10) – Correct

📄 **Explanation:** match and structural pattern matching were introduced in Python 3.10 via PEP 634.

Score: 1/1

Q3. Which is a valid match syntax?

✅ **Your Answer:** B – Correct

📄 **Explanation:** Option B uses the correct Python match-case syntax. Others are invalid.

Score: 1/1

Q4. What is a guard statement for?

✅ **Your Answer:** C – Correct

📄 **Explanation:** A guard (e.g. if $x > 5$) adds an extra condition inside a case.

Score: 1/1

Q5. Which case pattern matches any value?

✅ **Your Answer:** D – Correct

📄 **Explanation:** case `_` is the wildcard or default case. It matches everything not caught earlier.

Score: 1/1

■ **Section 1 Total: 5 / 5**

🧠 SECTION 2: STRUCTURED QUESTIONS (5 points total)

Q6. What's the output when value = 2?

✅ **Your Answer:** Two – Correct

📄 **Explanation:** Case 2 matches the value, so it prints "Two".

Score: 1/1

Q7. Explain how `_` works in match.

✓ **Your Answer:** Like else/default – Correct

📄 **Explanation:** `_` is the default case, same as else. Executes if no earlier case matches.

Score: 1/1

Q8. Role of case `x` if `x > 5`:

✓ **Your Answer:** Checks case + condition – Correct

📄 **Explanation:** You're right — it only matches if the value equals `x` **and** `x > 5` is also true.

Score: 1/1

Q9. Why use match instead of if-elif?

✓ **Your Answer:** Cleaner & shorter – Correct

📄 **Explanation:** Especially with many conditions or structured data, match is clearer and more elegant.

Score: 1/1

Q10. What happens if no case matches and no `_`?

✓ **Your Answer:** Nothing happens – Correct

📄 **Explanation:** If no match and no case `_`, Python just skips the block. No error.

Score: 1/1

■ **Section 2 Total: 5 / 5**

📖 SECTION 3: CODING QUESTIONS (10 points total)

Q11. Match day to weekday/weekend/invalid

✓ **Your Answer:** Correct match cases from 1–7 – Valid

📄 **Explanation:** You used all explicit case values with `_` fallback — it works perfectly.

Score: 2/2

Q12. Match "start" / "stop" command

✓ **Your Answer:** Correct logic – Slight redundancy

📄 **Feedback:** case "start" alone is enough. But still functional and acceptable.

Score: 2/2

Q13. Match tuple `(0, y)` and print "On Y-axis"

✗ **Your Answer:** Incorrect – did not match a tuple

📄 **Feedback:** You were trying to match a single value, not unpack a tuple. Correct way:

python

CopyEdit

match point:

```
case (0, y):
```

```
    print("On Y-axis")
```

Score: 0/2

Q14. Match list ["red", "blue"] and print "Has red"

✅ **Your Answer:** Works correctly

📝 **Feedback:** You matched colors and used if "red" in colors: — fully functional

Score: 2/2

Q15. Match number to check even/odd/negative/zero

✅ **Your Answer:** Fully correct logic

📝 **Feedback:** Excellent use of guards and order

Score: 2/2

■ **Section 3 Total: 8 / 10**

📊 **FINAL TOTAL: 18 / 20 (90%) – Excellent**

You're **very strong** in understanding the new Python 3.10+ match-case system.

Only miss was in tuple pattern matching, which is slightly advanced — but you're nearly there.

🏆 **Feedback Summary**

Question Group Score Comment

MCQs 5 / 5 Flawless

Structured 5 / 5 Strong answers with good reasoning

Coding 8 / 10 Only tuple matching needs revision

Final Score **18 / 20** 🏆 You passed with flying colors!

Let me know if you want a **follow-up exam** with more **advanced cases** like:

- Matching dicts, enums, or classes
- Deep tuple unpacking

- Nested match patterns

I'm ready anytime, bro.