CONDITIONAL STATEMENT IN JAVA

1. Explain the different types of conditional statements in Java.

Answer:

Java offers if, if-else, if-else if, and switch statements.

2. What are the differences between these statements?

Answer:

if: Executes code if a condition is true.

if-else: Executes code if true, else executes different code if false.

if-else if: Allows checking multiple conditions in sequence.

switch: Executes code based on the value of an integer or String expression.

3. What are common pitfalls to avoid in using conditionals?

Answer:

Nested conditionals leading to poor readability.

Not using else clauses when necessary.

Comparing objects by reference using == instead of equals().

Intermediate:

4. Describe the logical operators (&&, ||, !) and how they work.

Answer:

Explain AND, OR, NOT operations and their truth tables.

5. Give an example of using these operators in a conditional statement.

Answer:

Write code checking if a user is eligible for a discount based on age and purchase amount.

6. How does Java handle non-boolean expressions in conditionals?

Answer:

Any non-zero value is treated as true, and 0 is treated as false.

7. What are conditional expressions (?:) and when are they useful?

Answer:

Explain ternary operators as concise alternatives to simple if-else statements.

8. Discuss the advantages and disadvantages of using switch statements.

Answer:

Highlight readability for multiple conditions, but potential limitations for dynamic conditions or large numbers of cases.

9. In what scenarios would you consider using reflection to modify conditionals at runtime?

Answer:

Explain dynamic programming or configuration-based scenarios, but emphasize caution due to potential performance and maintainability impacts.

10. Write a function that takes a grade as input and returns a corresponding letter grade using conditionals.

Answer:

Demonstrate your understanding of conditionals, control flow, and logic development.

11. Explain how conditionals can be used for exception handling.

Answer:

Describe try-catch blocks and how conditionals within them control error handling behavior.