JAVA

Most Importand Interview Questions

1. Java Basics

1.What are the key features of Java?

Answer:

- **Platform Independent** – Write once, run anywhere (WORA).
- **Object-Oriented** – Everything in Java is based on objects.
- **Robust –** Has strong memory management and exception handling.
- **Secure** – No explicit pointers and uses JVM for security.
- **Multithreading** – Allows concurrent execution of multiple tasks.
- **Automatic Garbage Collection** – Manages memory automatically.

2. Why is Java called platform-independent?

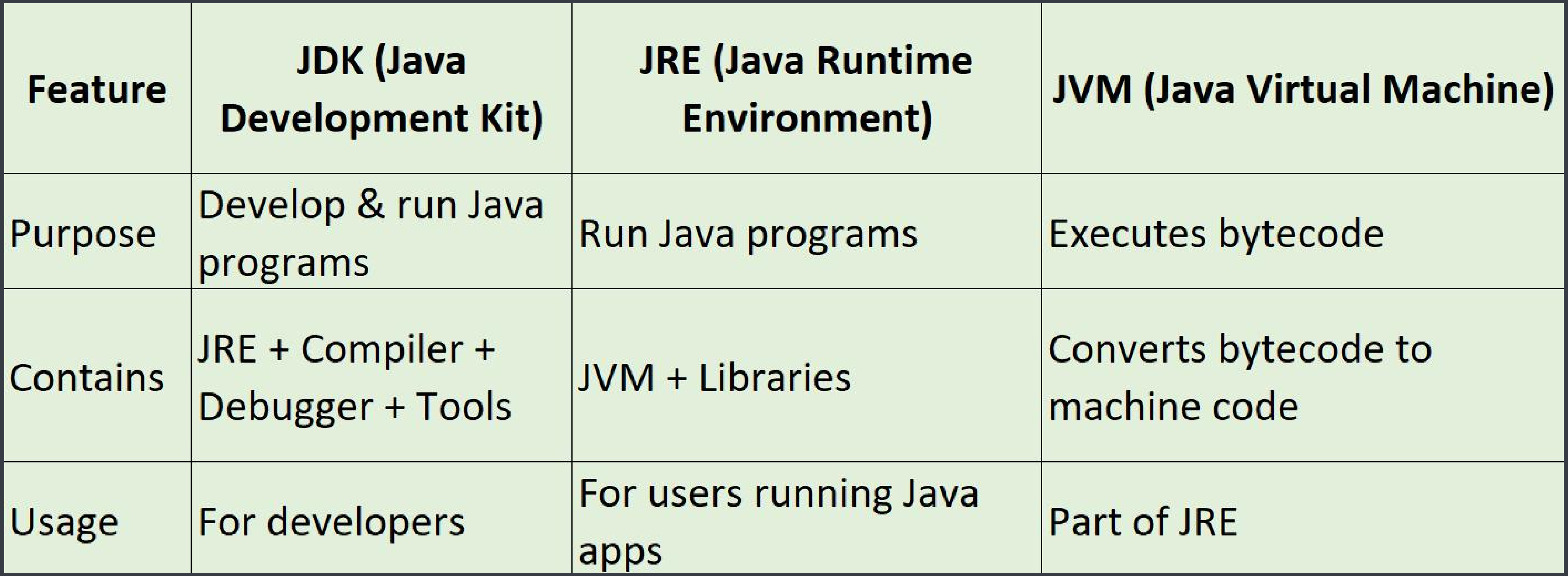
Answer:

- Java is **compiled into bytecode,** which runs on any Java Virtual Machine (JVM), making it platform-independent.

- Unlike C/C++, which compiles into **OS-specific machine code,** Java’s bytecode runs on Windows, Linux, Mac, etc., without modification.

3. Explain the difference between JDK, JRE, and JVM.

Answer:



Feature	JDK (Java Development Kit)	JRE (Java Runtime Environment)	JVM (Java Virtual Machine)
Purpose	Develop & run Java programs	Run Java programs	Executes bytecode
Contains	JRE + Compiler + Debugger + Tools	JVM + Libraries	Converts bytecode to machine code
Usage	For developers	For users running Java apps	Part of JRE

4. What is the difference between Java and C++?

Answer:

- Java is **platform-independent**, while C++ is platform-dependent.
- Java has **automatic garbage collection**, whereas C++ requires manual memory management.
- Java does **not use pointers** explicitly, while C++ does.

4. What is the difference between Java and C++?

Answer:

- Java is **platform-independent**, while C++ is platform-dependent.
- Java has **automatic garbage collection**, whereas C++ requires manual memory management.
- Java does **not use pointers** explicitly, while C++ does.

5. Why is Java not purely an object-oriented language?

Answer:

- Java uses **primitive data types (int, char, double, etc.)**, which are not objects. Since everything is not an object, **Java is not 100% object-oriented.**

2. Data Types & Variables

6. What are primitive data types in Java?

Answer:

Java has 8 primitive data types:

Integer Types: byte, short, int, long

Floating-Point Types: float, double

Character Type: char

Boolean Type: boolean (true or false)

7. Explain the difference between int and Integer.

Answer:

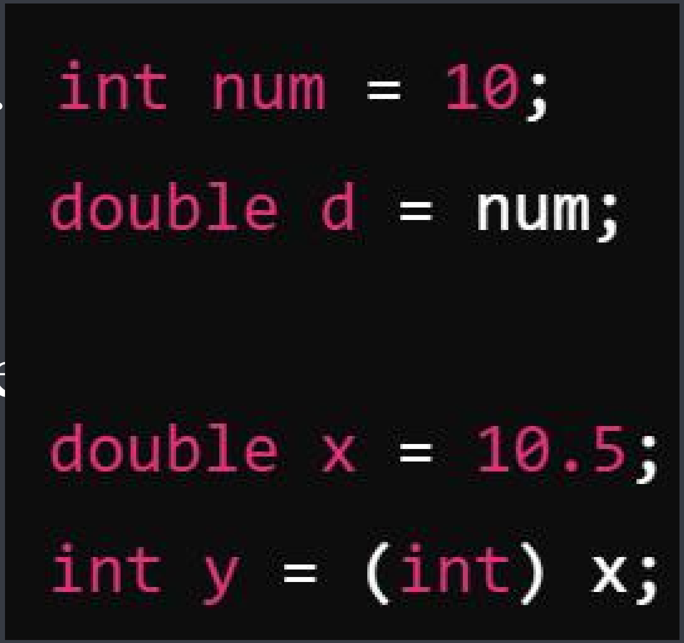
1. **int** is a primitive data type.
2. **Integer** is a wrapper class in Java (java.lang.Integer) that provides methods for operations like parsing, comparing, and converting.

8. What is type casting in Java? Give an example.

Answer:

Type casting converts one data type to another.

1. **Implicit Casting (Widening)** – Smaller to larger type (automatic).
2. **Explicit Casting (Narrowing)** – Larger to smalle type (manual).



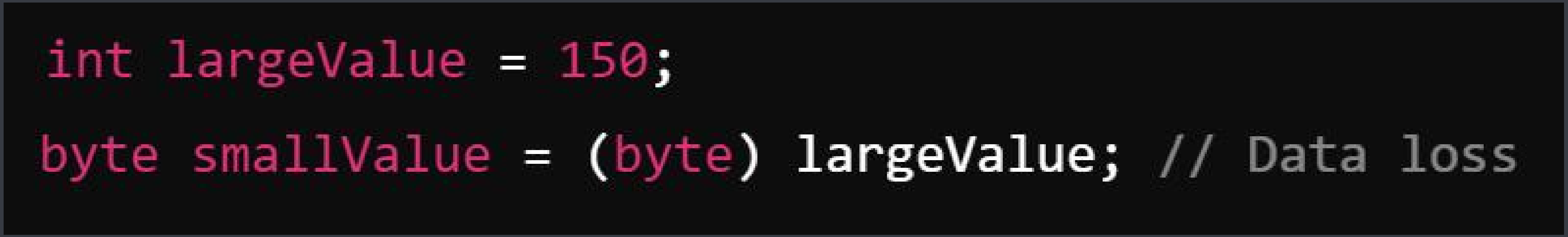
```
int num = 10;
double d = num;

double x = 10.5;
int y = (int) x;
```

9. What happens if you store a larger data type into a smaller one?

Answer:

This results in data loss.



```
int largeValue = 150;
byte smallValue = (byte) largeValue; // Data loss
```

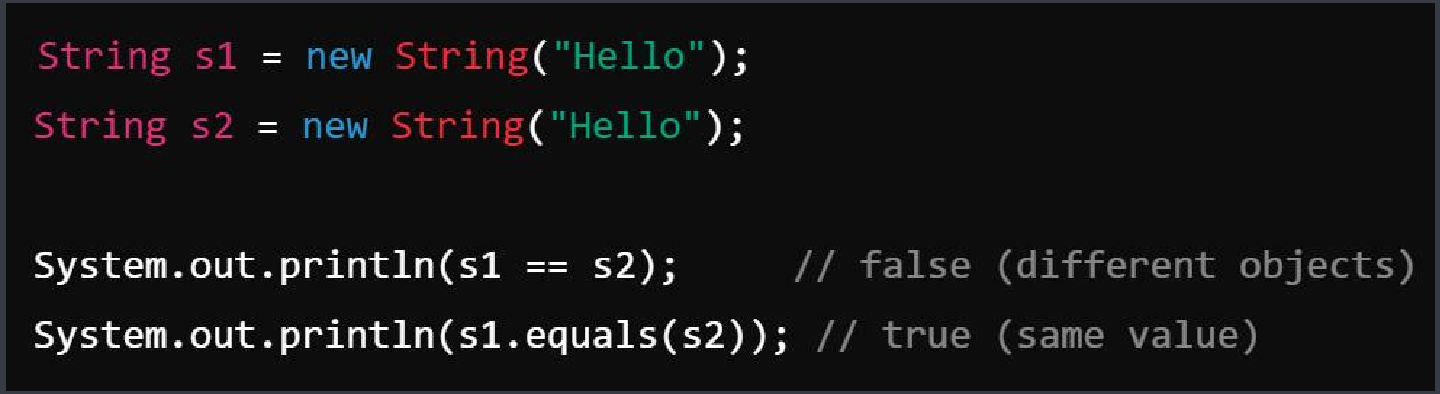
If largeValue is outside byte range (-128 to 127), incorrect data is stored.

3. Operators & Conditional Statements

10. What is the difference between == and .equals() in Java?

Answer:

1. == compares references (memory locations).
2. .equals() compares actual values.



```java
String s1 = new String("Hello");
String s2 = new String("Hello");

System.out.println(s1 == s2);     // false (different objects)
System.out.println(s1.equals(s2)); // true (same value)
```

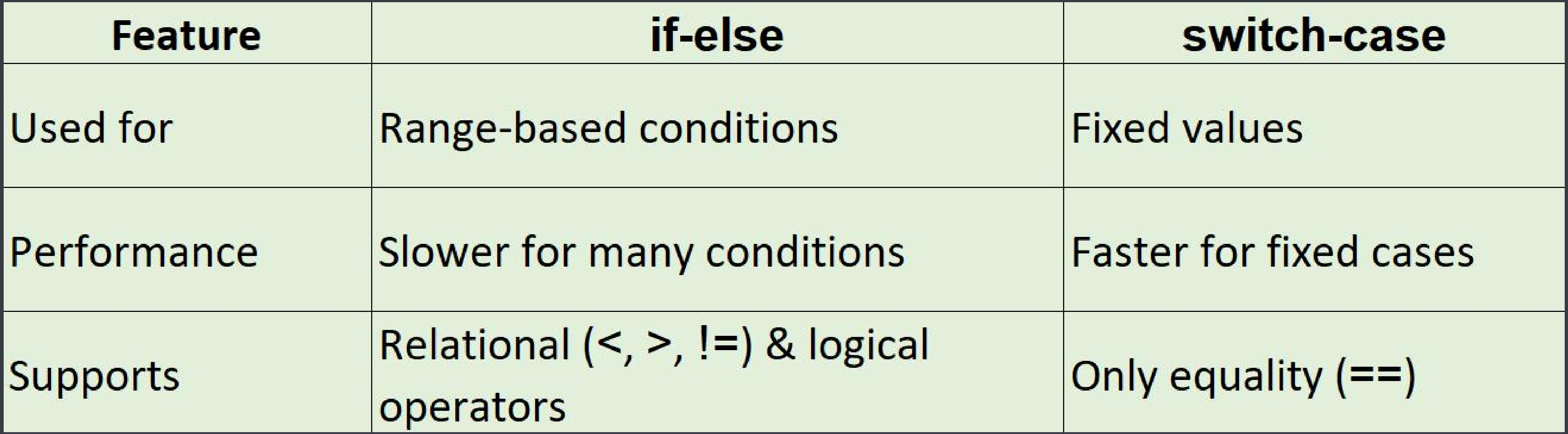
11. Explain the difference between && and & in Java.

Answer:

1. **&& (Logical AND)** – Short-circuiting occurs (stops if first condition is false).
2. **& (Bitwise AND)** – Always evaluates both conditions.

12. What is the difference between if-else and switch-case?

Answer:



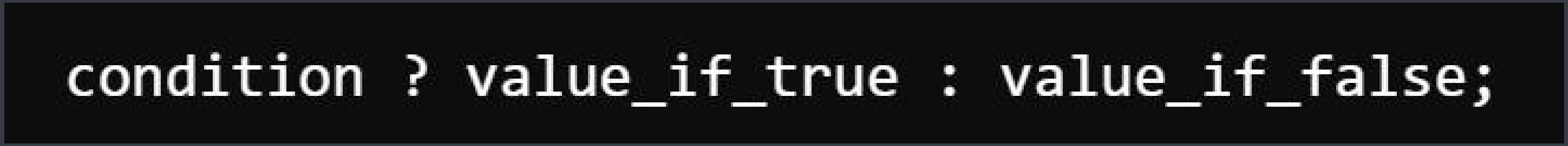
Feature	if-else	switch-case
Used for	Range-based conditions	Fixed values
Performance	Slower for many conditions	Faster for fixed cases
Supports	Relational (<, >, !=) & logical operators	Only equality (==)

13. What is the use of the ternary operator?

Answer:

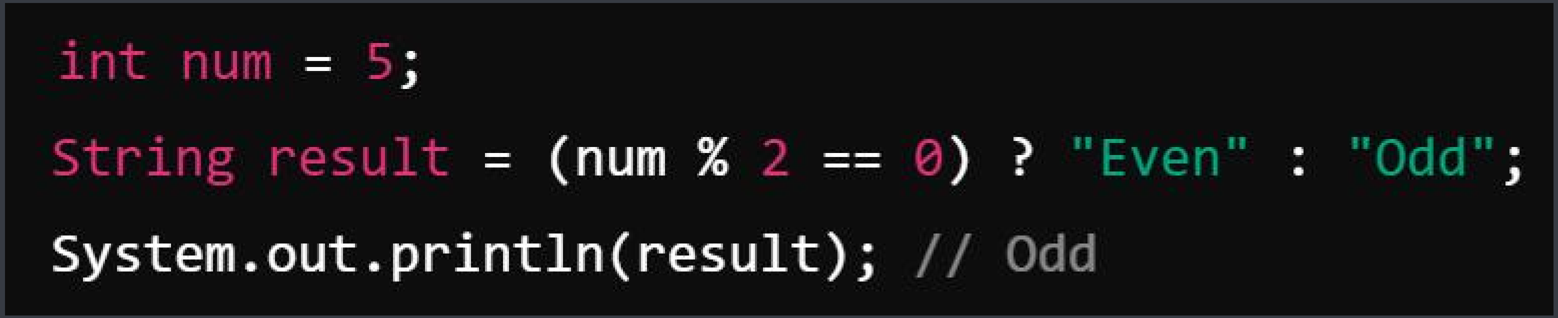
1. A shorthand for if-else conditions.

Syntax:



```
condition ? value_if_true : value_if_false;
```

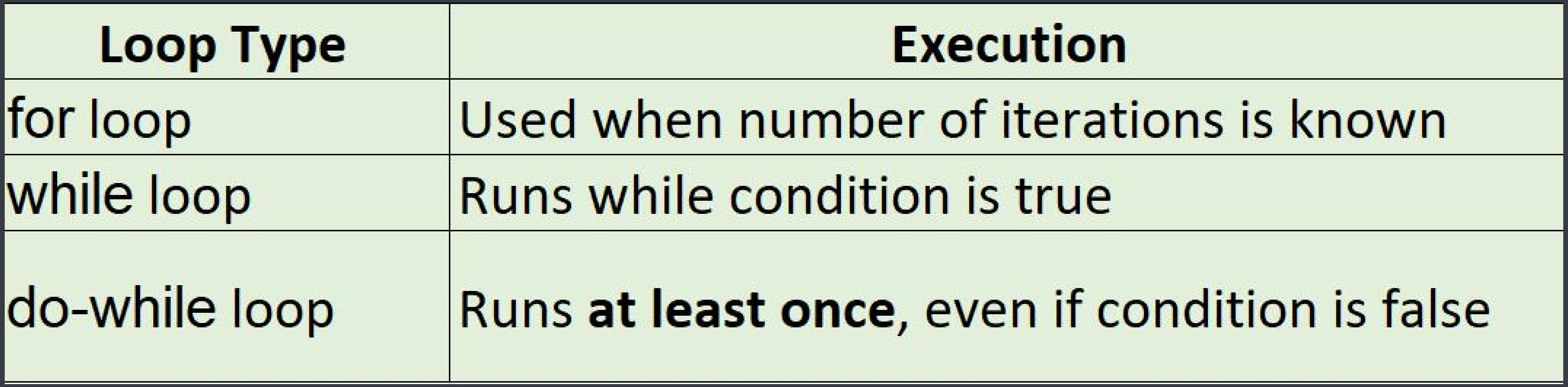
Example:



```
int num = 5;
String result = (num % 2 == 0) ? "Even" : "Odd";
System.out.println(result); // Odd
```

14. Explain the difference between for, while, and do-while loops.

Answer:



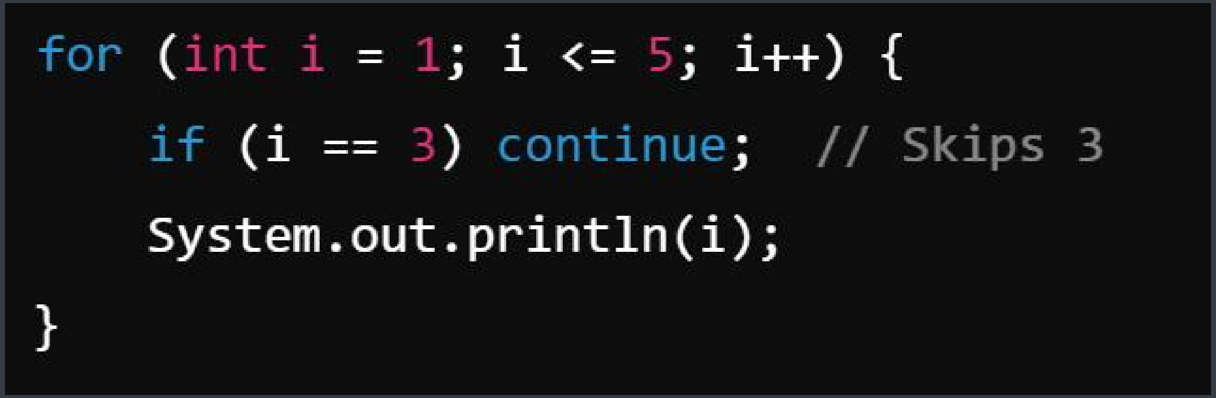
Loop Type	Execution
for loop	Used when number of iterations is known
while loop	Runs while condition is true
do-while loop	Runs **at least once**, even if condition is false

15. What is the use of break and continue in loops?

Answer:

1. **break;** exits the loop completely.

2. **continue;** skips the current iteration and moves to the next one.



```
for (int i = 1; i <= 5; i++) {
    if (i == 3) continue;  // Skips 3
    System.out.println(i);
}
```

Output: 1 2 4 5

THANK YOU

Thank You