Default and Static Methods in Java

1. Default Methods

- Introduced in Java 8.
- Declared with the **default** keyword inside an interface.
- Provide a default implementation so that implementing classes automatically inherit it.
- Helps add new methods to interfaces without breaking old code.

Example:

```
interface Vehicle {
void start();

default void stop() {
System.out.println("Vehicle stopped");
}
}

class Car implements Vehicle {
public void start() {
System.out.println("Car started");
}
}

public class Main {
public static void main(String[] args) {
Car car = new Car();
car.start(); // Car started
car.stop(); // Vehicle stopped (default method used)
}
}
```

2. Static Methods

- Introduced in Java 8.
- Declared with the **static** keyword inside an interface.
- Belong to the interface itself, not the implementing classes.
- Cannot be overridden by implementing classes.
- Useful for utility/helper methods inside interfaces.

Example:

```
interface Vehicle {
void start();

static void service() {
System.out.println("Vehicle is being serviced");
}
}

class Car implements Vehicle {
public void start() {
System.out.println("Car started");
}
}

public class Main {
public static void main(String[] args) {
Car car = new Car();
car.start(); // Car started
Vehicle.service(); // Vehicle is being serviced (static method call)
}
}
```

3. Quick Comparison

Feature	Default Method	Static Method
Keyword	default	static
Location	Inside interface	Inside interface
Belongs to	Implementing class (inherited)	Interface itself
Can be overridden?	Yes	No
Call using	Object of implementing class	Interface name
Use case	Provide backward-compatible implementation	Utility/helper methods