SNS COLLEGE OF TECHNOLOGY

Kurumbapalayam (Po), Coimbatore – 641 107

AN AUTONOMOUS INSTITUTION

Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF COMPUTER SCIENCE AND TECHNOLOGY

23CSB201-OBJECT ORIENTED PROGRAMMING

UNIT IV - EXCEPTION AND MULTITHREADING

TOPIC 6 -Thread Creation

1. What is Thread Creation?

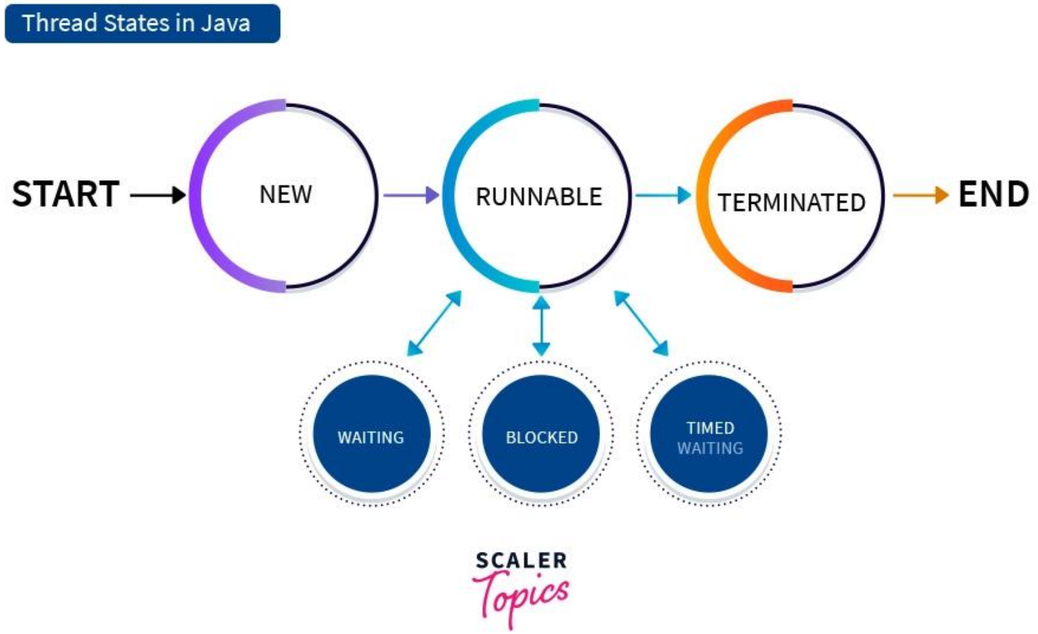
- **Thread creation** means creating a **new path of execution** in a program so that multiple tasks can run simultaneously.
- In Java, threads are created using the classes and interfaces from the java.lang **package**

2. Ways to Create Threads in Java

There are **two main ways**:

✓ **1. Extending the Thread class**

✓ **2. Implementing the Runnable interface**

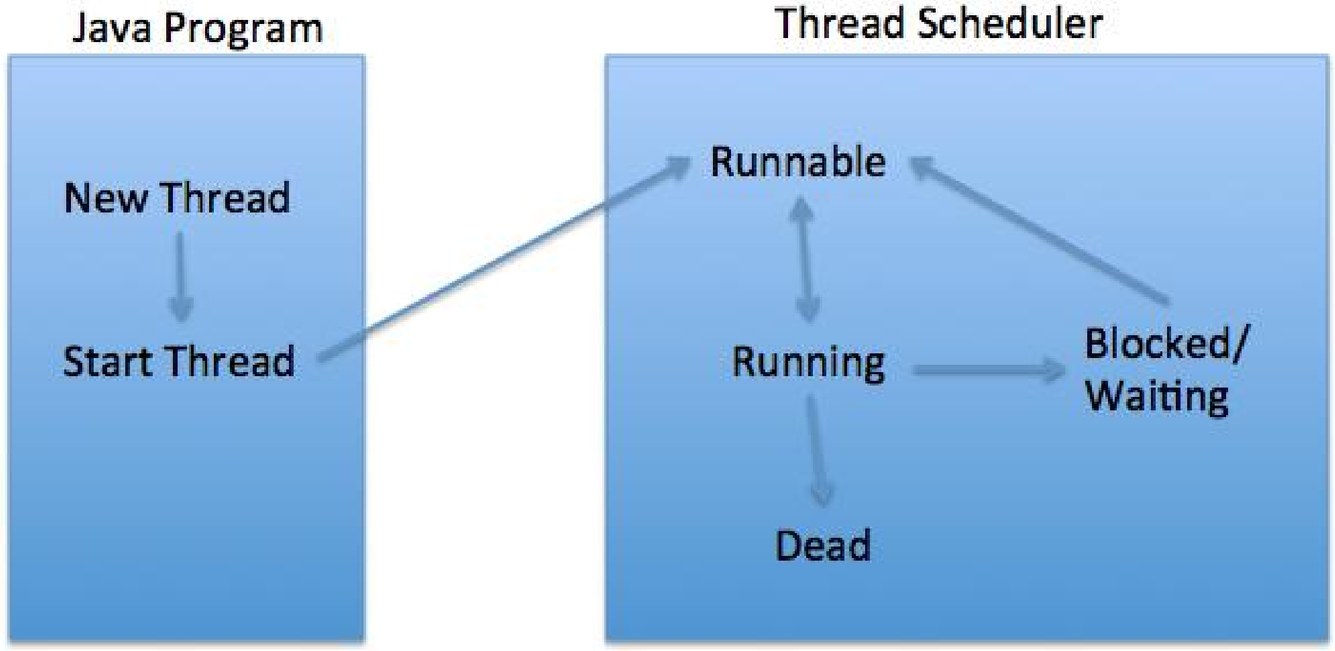


3. Method 1: Extending Thread Class

- In this method, you create a class that **extends the** Thread **class** and override the run() method.

Steps:

1. Extend Thread class
2. Override run() method
3. Create object of the class
4. Call start() method



Example:

```
class MyThread extends Thread {
    public void run() {
        System.out.println("Thread is running...");
    }
    public static void main(String[] args) {
        MyThread t = new MyThread();
        t.start();  // starts new thread
    }
}
```

Output:

Thread is running...

Important Note:

Do NOT call run() **directly**

t.run(); // wrong

✓ This will not create a new thread

✓ It behaves like a normal method call

4. Method 2: Implementing Runnable Interface

This is the **most recommended method** in Java.

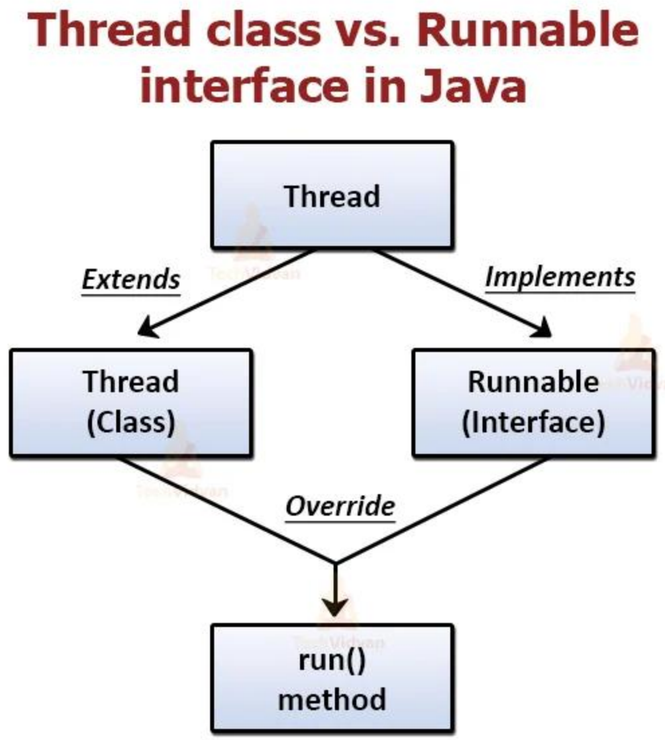
Steps:

1.Implement Runnable interface

2.Override run() method

3.Pass object to Thread class

4.Call start()



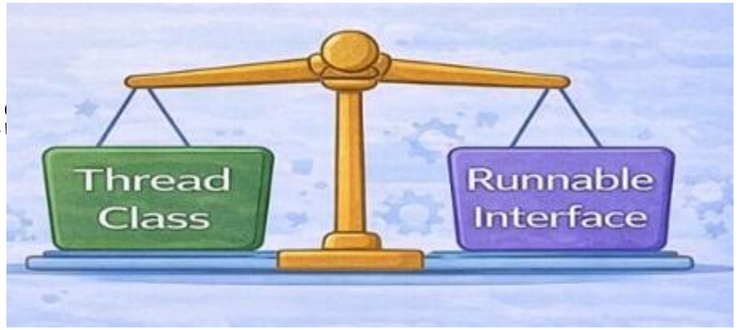
Example:

```
class MyRunnable implements Runnable {
    public void run() {
        System.out.println("Thread is running...");
    }
    public static void main(String[] args) {
        MyRunnable obj = new MyRunnable();
        Thread t = new Thread(obj);
        t.start();
    }
}
```

Output:

```
Thread is running...
```

5. Difference Between Two Methods



Feature	Thread Class	Runnable Interface
Inheritance	Uses inheritance	Uses interface
Flexibility	Less flexible	More flexible
Multiple inheritance	Not possible	Possible
Recommended	❌ No	✅ Yes

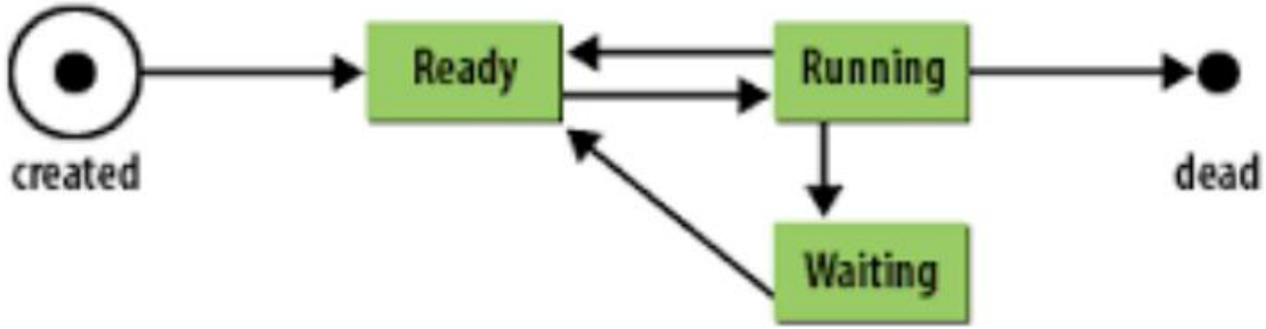
6. Thread Class vs Runnable (Concept)

Java does **not support multiple inheritance**

So if you extend Thread, you cannot extend any other class

But using Runnable, you can:

class A extends B imp



7. Creating Multiple Threads

```
class MyThread extends Thread {
    public void run() {
        System.out.println(Thread.currentThread().getName());
    }
    public static void main(String[] args) {
        MyThread t1 = new MyThread();
        MyThread t2 = new MyThread();
        t1.start();
        t2.start();
    }
}
```

8. Thread Naming

```
t1.setName("Thread-1");
System.out.println(t1.getName());
```

9. Using Lambda (Modern Way - Java 8+)

```
public class Main {
    public static void main(String[] args) {
        Thread t = new Thread(() -> {
            System.out.println("Thread using lambda");
        });
        t.start();
    }
}
```

10. Thread Life After Creation

After calling start():

👉Thread goes to **Runnable state**

👉Then scheduler moves it to **Running state**

11. Common Mistakes

❌ Calling run() instead of start()

❌ Not overriding run()

❌ Forgetting to handle exceptions

12. Real-Time Example

```
class Download extends Thread {
    public void run() {
        System.out.println("Downloading file...");
    }
}
class Music extends Thread {
    public void run() {
        System.out.println("Playing music...");
    }
}
public class Main {
    public static void main(String[] args) {
        Download d = new Download();
        Music m = new Music();
        d.start();
        m.start();
    }
}
```

13. Advantages of Thread Creation

✓ Parallel execution

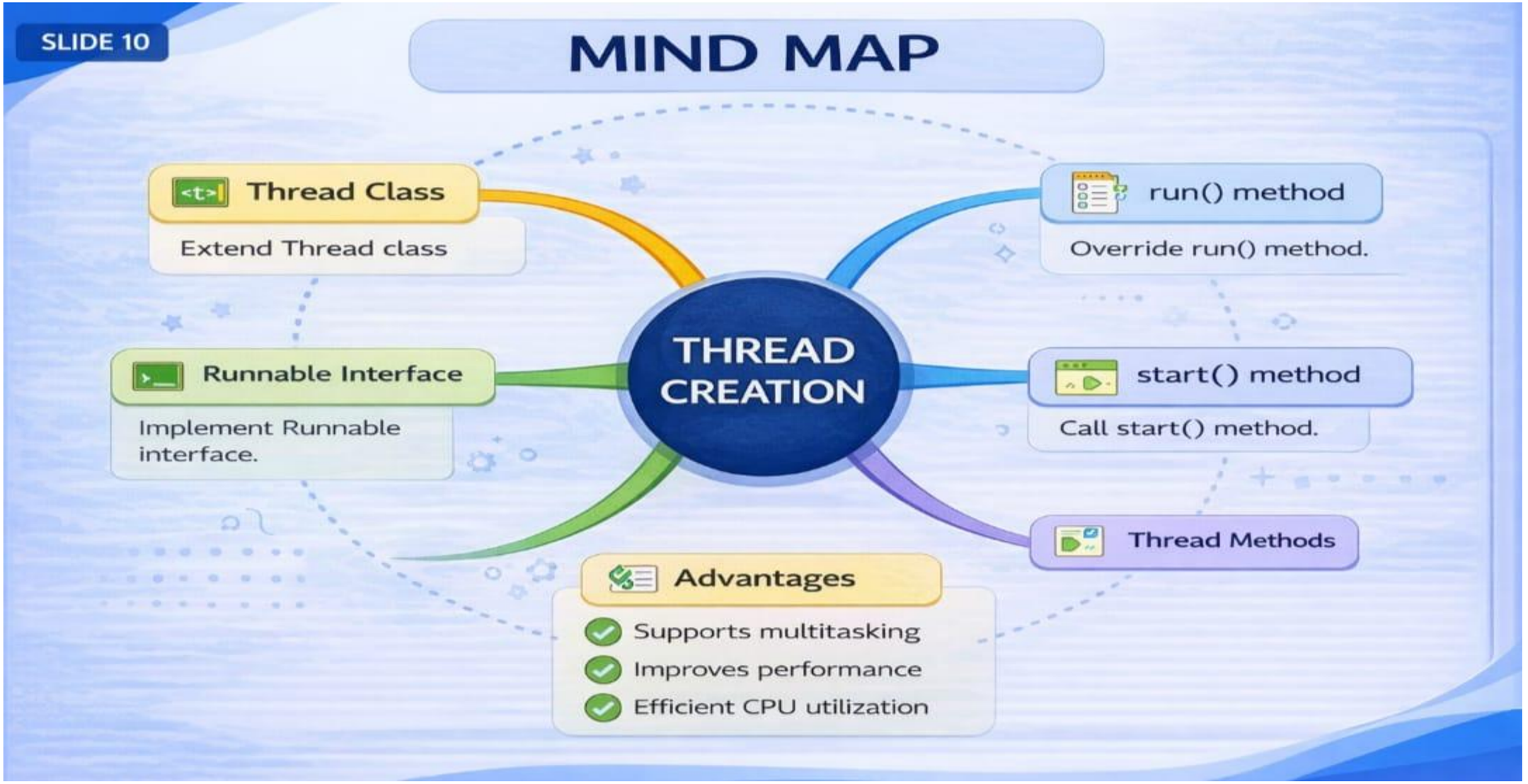
✓ Faster processing

✓ Better CPU utilization

✓ Responsive applications

14. Summary

- Threads are created using **Thread class** or **Runnable interface**
- start() method is used to begin execution
- run() contains the logic
- Runnable is **preferred method**


SLIDE 10
MIND MAP
THREAD CREATION
Thread Class
Extend Thread class
Runnable Interface
Implement Runnable interface.
run() method
Override run() method.
start() method
Call start() method.
Thread Methods
Advantages
Supports multitasking
Improves performance
Efficient CPU utilization

Assessment

1. What happens if you call run() instead of start()?
2. Can you start a thread twice?
3. What is a Daemon Thread?
4. What is the difference between Runnable and Callable?

THANK YOU