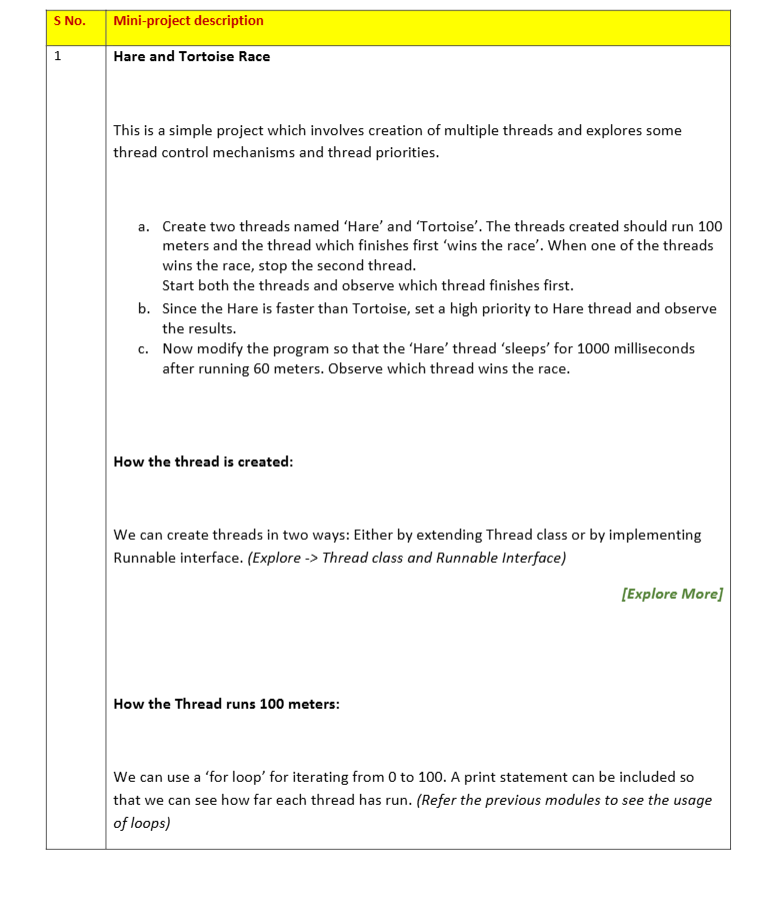
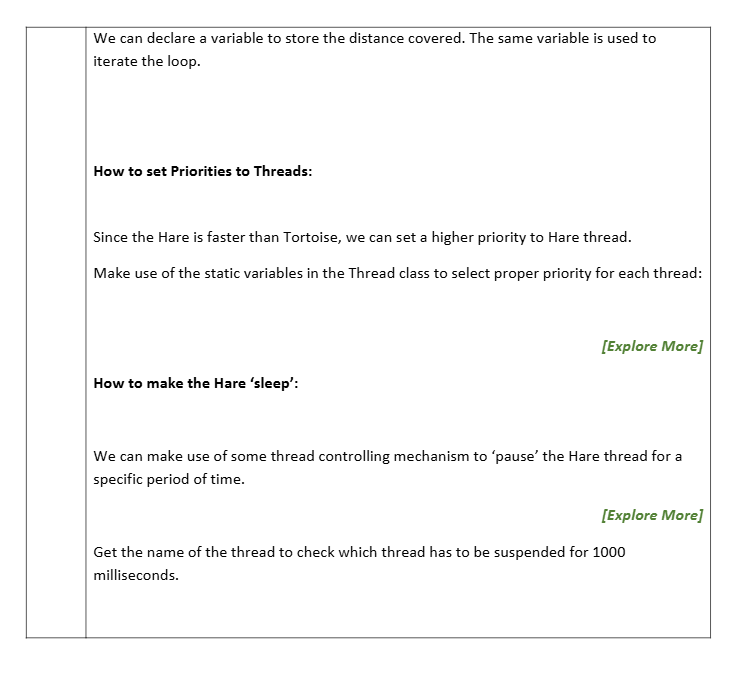
# MINI PROJECT

**Multiplethredings**

**BYAGARI PAVAN**

**PBL\_ID: J\_251890123**

****

****

**Program:**

class RaceThread extends Thread {

private String name;

RaceThread(String name) {

this.name = name;

}

@Override

public void run() {

for (int i = 1; i <= 100; i++) {

System.out.println(name + " ran " + i + " meters");

// Hare sleeps after 60 meters

if (name.equals("Hare") && i == 60) {

System.out.println("Hare is tired and sleeping for 1 second...");

try {

Thread.sleep(1000); // Hare sleeps for 1 second

} catch (InterruptedException e) {

e.printStackTrace();

}

}

try {

// Hare runs faster (10ms), Tortoise runs slower (50ms)

if (name.equals("Hare")) {

Thread.sleep(10);

} else {

Thread.sleep(50);

}

} catch (InterruptedException e) {

e.printStackTrace();

}

}

System.out.println("🏁 " + name + " has finished the race!");

}

}

public class HareTortoiseRace {

public static void main(String[] args) {

RaceThread hare = new RaceThread("Hare");

RaceThread tortoise = new RaceThread("Tortoise");

// Setting priority: Hare gets higher priority

hare.setPriority(Thread.MAX\_PRIORITY);

tortoise.setPriority(Thread.MIN\_PRIORITY);

// Start both threads

hare.start();

tortoise.start();

}

}

**OUTPUT:-**

Hare ran 1 meters

Tortoise ran 1 meters

Hare ran 2 meters

Tortoise ran 2 meters

...

Hare ran 60 meters

Hare is tired and sleeping for 1 second...

Tortoise ran 60 meters

Tortoise ran 61 meters

...

Tortoise has finished the race!

Hare has finished the race!