

Vijaya Lakshmi Sravanti

1. Write a program that creates two threads. Each Thread should print its thread ID (TID) and a unique message to the console. Ensure that the output from both threads is interleaved.

```
package vijaya;

public class vijaya implements Runnable {

    private String message;
    public vijaya(String message) {
        this.message = message;
    }
    public void run() {
        for (int i = 0; i < 4; i++) {

            System.out.println(Thread.currentThread().getId() + ": "
+ message);

            try {
                Thread.sleep(50); // Optional delay to increase
interleaving chances
            } catch (Exception e) {
                System.out.println(e);
            }
        }
    }
}

package vijaya;

public class Interleaved {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Thread thread1 = new Thread(new vijaya("Thread 1"));
        Thread thread2 = new Thread(new vijaya("Thread 2"));
        thread1.start();
        thread2.start();
    }
}
```

Output:

```
21[TID]: Thread 1
22[TID]: Thread 2
22[TID]: Thread 2
21[TID]: Thread 1
21[TID]: Thread 1
22[TID]: Thread 2
21[TID]: Thread 1
22[TID]: Thread 2
Thread 2
```

2. Write a program that creates multiple threads with different priorities. Observe how the operating system schedules threads with different priorities and explain the results.

```
package vijaya;

public class vijaya implements Runnable {

    public void run() {
        for (int i = 0; i < 4; i++) {

            System.out.println(Thread.currentThread().getName() + ":
Priority "
                                +
Thread.currentThread().getPriority() + ", Count: " + i);
            try {
                Thread.sleep(50); // Optional delay to increase
interleaving chances
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    }
}
```

```

package vijaya;

public class Interleaved {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Thread Thread1 = new Thread(new vijaya(), "Low
Priority Thread");
        Thread Thread2 = new Thread(new vijaya(), "Normal
Priority Thread");
        Thread Thread3 = new Thread(new vijaya(), "High
Priority Thread");
        // Set thread priorities
        Thread1.setPriority(Thread.MIN_PRIORITY);
        Thread2.setPriority(Thread.NORM_PRIORITY);
        Thread3.setPriority(Thread.MAX_PRIORITY);
        Thread1.start();
        Thread2.start();
        Thread3.start();
    }

}

```

Output:

```

Normal Priority Thread: Priority 5, Count: 0
Low Priority Thread: Priority 1, Count: 0
High Priority Thread: Priority 10, Count: 0
High Priority Thread: Priority 10, Count: 1
Low Priority Thread: Priority 1, Count: 1
Normal Priority Thread: Priority 5, Count: 1
High Priority Thread: Priority 10, Count: 2
Low Priority Thread: Priority 1, Count: 2
Normal Priority Thread: Priority 5, Count: 2
High Priority Thread: Priority 10, Count: 3
Normal Priority Thread: Priority 5, Count: 3
Low Priority Thread: Priority 1, Count: 3

```

3. Write a Java program that creates two threads and prints "Thread A" from the first thread and "Thread B" from the second thread. Make sure both threads run concurrently.

```

package vijaya;

public class vijaya implements Runnable {

```

```

        private String message;
        public vijaya(String message) {
            this.message = message;
        }
        public void run() {
            for (int i = 0; i < 4; i++) {
                System.out.println(message);

                try {
                    Thread.sleep(50); // Optional delay to increase
interleaving chances
                } catch (Exception e) {
                    System.out.println(e);
                }
            }
        }
    }
}

```

```
package vijaya;
```

```

public class Interleaved {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Thread thread1 = new Thread(new vijaya("Thread 1"));
        Thread thread2 = new Thread(new vijaya("Thread 2"));
        thread1.start();
        thread2.start();
    }

}

```

Output:

```

Thread 1
Thread 2
Thread 2
Thread 1
Thread 1
Thread 2
Thread 1
Thread 2

```

