**Threads**

In Java, a thread is the smallest unit of execution within a process. It allows a program to perform multiple tasks simultaneously, effectively enabling multitasking or concurrent execution. Threads are independent sequences of instructions that can run concurrently with other threads, allowing for efficient utilization of system resources and better responsiveness in applications.

**Thread Class and Runnable Interface:**

In Java, you can create a thread by extending the Thread class or implementing the Runnable interface. Extending the Thread class involves overriding its run() method, where you define the code to be executed by the thread. Implementing the Runnable interface requires implementing the run() method as well, and then passing an instance of the Runnable to a Thread object.

**Thread Class:**

**class** Multi **extends** Thread{

**public** **void** run(){

System.out.println("thread is running...");

}

**public** **static** **void** main(String args[]){

Multi t1=**new** Multi();

t1.start();

 }

}

**Runnable Interface**

**class** Multi3 **implements** Runnable{

**public** **void** run(){

System.out.println("thread is running...");

}

**public** **static** **void** main(String args[]){

Multi3 m1=**new** Multi3();

Thread t1 =**new** Thread(m1);   // Using the constructor Thread(Runnable r)

t1.start();

 }

}