```java
package Week_06;

public class MultithreadingTask implements Runnable {
    int threadNumber;


    public MultithreadingTask(int threadNumber) {

        this.threadNumber = threadNumber;
    }

    @Override
    public void run() {
        for (int i = 1; i <= 3; i++) {
            System.out.println(Thread.currentThread().getName() + ": " +
threadNumber + " * " + i + " = " + (threadNumber * i));
            try{
                Thread.sleep((long) (Math.random()*1000));
            }catch (InterruptedException e){
                e.printStackTrace();
            }
        }
    }
}
public class MultiplicationThread {
    public static void main(String[] args) {
        Thread thread1 = new Thread(new MultithreadingTask(1));
        thread1.setName("Thread-0");
        Thread thread2 = new Thread(new MultithreadingTask(3));
        thread2.setName("Thread-1");
        Thread thread3 = new Thread(new MultithreadingTask(2));
        thread3.setName("Thread-2");

        thread1.start();
        thread2.start();
        thread3.start();


    }
}
```

output:

Thread-0: 1 * 1 = 1

Thread-2: 2 * 1 = 2

Thread-2: 3 * 1 = 3

Thread-2: 2 * 2 = 4

Thread-2: 2 * 3 = 6

Thread-0: 1 * 2 = 2

Thread-2: 3 * 2 = 6

Thread-0: 1 * 3 = 3

Thread-2: 3 * 3 = 9

```java
package Week_06;

public class MyThreads extends Thread {
    private volatile boolean running = true;
    public void run() {
        while (running) {
            System.out.println("Please wait while thread is running...");
            try{
                sleep(2000);
            }catch(InterruptedException e){
                Thread.currentThread().interrupt();
                System.out.println("Thread is interrupted");
            }
        }
        System.out.println("Thread is interrupt,thread is shutting down");
    }
    public void shutDown() {
        running = false;
    }
package Week_06;

public class MyVolatile {
    public static void main (String[] args) {
        MyThreads myThreads = new MyThreads();
        myThreads.start();
        System.out.println("Press ENTER to shutdown the thread");
        try{
            System.in.read();

        }catch(Exception e){
            e.printStackTrace();
        }
        myThreads.shutDown();
    }
}
```

Output:

Press ENTER to shutdown the thread

Please wait while thread is running...

Please wait while thread is running...

Please wait while thread is running...

Please wait while thread is running...

Please wait while thread is running...

Please wait while thread is running...

Please wait while thread is running...

Please wait while thread is running...

Please wait while thread is running...

Thread is interrupt,thread is shutting down

Process finished with exit code 0