

MULTI THREADED APPLICATION

AIM:

To write a program that implements a multi-threaded application that has three threads. First thread generates a random integer every 1 second and if the value is even, second thread computes the square of the number and prints. If the value is odd, the third thread will print the value of cube of the number.

ALGORITHM:

1. Import the java packages.
2. Create a thread that generates random number, Obtain one random number and check is odd or even.
3. If number is even then create and start thread that computes square of a number, Compute number * number and display the answer.
4. Notify to Random number thread and goto step 7.
5. If number is odd then create and start thread that computes cube of a number, Compute number * number * number and display the answer.
6. Notify to Random number thread and goto step 7.
7. Wait for 1 Second and Continue to Step 3 until user wants to exits.

PROGRAM:

*//File Name should be **Multithread.java***

```
import java.util.*;
```

```
class Even implements Runnable
```

```
{ public int x;
```

```
public Even(int x)
```

```
{ this.x = x;
```

```
}
```

```
public void run() {
```

```
System.out.println("New Thread "+ x +" is EVEN and Square of "+ x + " is: " + x * x);
```

```
}
```

```
}
```

```

class Odd implements Runnable
{ public int x;
  public Odd(int x) {
    this.x = x; }
  public void run() {
    System.out.println("New Thread "+ x +" is ODD and Cube of " + x + " is: " + x * x * x);
  }
}

```

```

class Generate extends Thread
{
  public void run()
  {
    int num = 0;
    Random r = new Random();
    try {
      for (int i = 0; i < 5; i++)
      {
        num = r.nextInt(100);
        System.out.println("Main Thread Generates Random Integer: " + num); if
        (num % 2 == 0)
        {
          Thread t1 = new Thread(new Even(num));
          t1.start(); } else
          {
            Thread t2 = new Thread(new Odd(num)); t2.start();
          }
          Thread.sleep(1000);
          System.out.println("-----");
        } }
      catch (Exception ex)
      {
        System.out.println(ex.getMessage());
      }
    }
  }
}

```

```

public class Multithread
{

```

```
public static void main(String[] args)
{
    Generate g = new Generate(); g.start();
}
}
```

NOTE:

To Compile:

javac Multithread.java To

Run:

java Multithread

OUTPUT:

RESULT:

Thus the Implementation for application for multithreading has been successfully executed.