Creating Threads in Java

every Java program has at least one thread called main thread. When a program starts, main thread starts running immediately.

Apart from this main thread, we can also create our own threads in a program that is called child thread. Every child threads create from its main thread known as parent thread.

There are two ways to create a new thread in Java. They are as follows:

1. One is by extending java.lang.Thread class

2. Another is by implementing java.lang.Runnable interfac

```
// Custom thread class.
public class MyThread extends Thread
{
// Override the run method in Runnable.
 public void run()
 {
   System.out.println("New thread running ");
 }
 public static void main(String[] args)
 {
  System.out.println("Main thread running");

// Create an object of MyThread class.
   MyThread th = new MyThread();

// Create an object of Thread class and pass the object reference variable th.
   Thread t = new Thread(th);

// Now run thread on the object. For this,
//call start() method using reference variable t.
   t.start(); // This thread will execute statements inside run()
// method of MyThread object.
 }
}
```

check when a new thread creates on the basis of the active number of threads.

```
public class NewThread extends Thread {
public  void run()
{
 Thread th1 = Thread.currentThread();
 System.out.println(th1);
  System.out.println("New thread strats running");
  System.out.println("I am in run() method");
}
public static void main(String[] args)
{
 System.out.println("Main thread starts running");
 Thread ct1 = Thread.currentThread();
 System.out.println(ct1);
```

```
   int ac1 = Thread.activeCount();
   System.out.println(ac1);
------------------------------
// Create an object of NewThread class.
   NewThread nt = new NewThread();
   int ac2 = Thread.activeCount();
   System.out.println(ac2);
----------------------------------
// Create an object of Thread class and pass the object reference variable nt.
   Thread t = new Thread(nt);
   int ac3 = Thread.activeCount();
   System.out.println(ac3);
-----------------------------
// Now run thread on the object. For this,
//call start() method using reference variable t.
   t.start(); // This thread will execute statements inside run() method of
//NewThread object.
   int ac4 = Thread.activeCount();
   System.out.println(ac4);
   t.setName("NewThread"); // Setting a new name of thread.
 }
}
```

Creating Threads in Java using Runnable Interface

```java
public class MyThread implements Runnable
{
 public void run()
 {
  System.out.println("New thread running ");
  for(int i = 1; i <= 5; i++)
  {
   System.out.println(i);
  }
 System.out.println(Thread.currentThread());
}
public static void main(String[] args)
{
 System.out.println("Main thread running");

// Create an object of MyThread class.
   MyThread th = new MyThread();

// Create an object of Thread class and pass reference variable th to Thread class constructor.
   Thread t = new Thread(th);

   t.start(); // This thread will execute statements inside run() method of MyThread object.
 }
}
```

Multitasking with a single Thread in Java

```java
public class MyThread implements Runnable
{
 int a = 20, b = 10;
public void run()
{
 addition(); // task1
```

```java
  subtraction(); // task2
  multiplication(); // task3
 }
void addition()
{
 int sum = a + b;
 System.out.println("Addition of two numbers: " +sum);
}
void subtraction()
{
 int sub = a - b;
 System.out.println("Subtraction of two numbers: " +sub);
}
void multiplication()
{
 int multiply = a * b;
 System.out.println("Multiplication of two numbers: " +multiply);
}
public static void main(String[] args)
{
 System.out.println("Main thread running");

  MyThread th = new MyThread();
  Thread t = new Thread(th);
  t.start(); // This thread will execute statements inside run()
//method of MyThread object.
  }
}
```