**Develop a program to create a class MyThread in this class a constructor, call the base class**

**constructor, using super and start the thread. The run method of the class starts after this. It can**

**be observed that both main thread and created child thread are executed concurrently.**

**Save Filename as: ThreadConstructor.java**

**Solution:-**

**class MyThread extends Thread**

**{**

**MyThread (String name)**

**{**

**super (name); // Call the base class constructor with threadname**

**start (); // Start the thread**

**}**

**public void run ()**

**{**

**try**

**{**

**for (int i = 5; i > 0; i--)**

**{**

**System.out.println (getName () + ": " + i);**

**Thread.sleep (1000);**

**}**

**}**

**catch (InterruptedException e)**

**{**

**System.out.println (getName () + " interrupted.");**

**}**

**System.out.println (getName () + " exiting.");**

**}**

**}**

**public class ThreadConstructor**

**{**

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

**{**

**new MyThread ("Child Thread");**

**try**

**{**

**for (int i = 5; i > 0; i--)**

**{**

**System.out.println ("Main Thread: " + i);**

**Thread.sleep (2000);**

**}**

**}**

**catch (InterruptedException e)**

**{**

**System.out.println ("Main Thread interrupted.");**

**}**

**System.out.println ("Main Thread exiting.");**

**}**

**}**

**Compile As: javacThreadConstructor.java**

**Run As: java ThreadConstructor**

**Output:**

**Main Thread: 5**

**Child Thread: 5**

**Child Thread: 4**

**Main Thread: 4**

**Child Thread: 3**

**Child Thread: 2**

**Main Thread: 3**

**Child Thread: 1**

**Child Thread exiting.**

**Main Thread: 2**

**Main Thread: 1**

**Main Thread exiting**