INTCDE21ID008

STAGE-3

916483 - Naveen S

Day 3 – C# Additional Topics Async Programming, Multithreading

Hands-On 2:

Multithreading - ThreadStart

Printer.cs

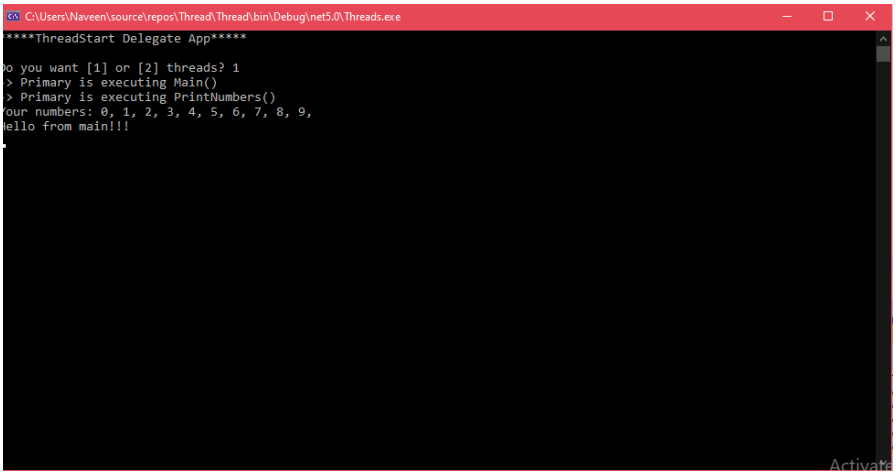
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading;
using System.Threading.Tasks;

namespace Threads
{
  public  class Printer
    {
        public void PrintNumbers()
        {

            Console.WriteLine("-> {0} is executing PrintNumbers()",
Thread.CurrentThread.Name);
            Console.Write("Your numbers: ");
            for (int i = 0; i < 10; i++)
            {
                Console.Write("{0}, ", i);
                Thread.Sleep(2000);
            }
            Console.WriteLine();
        }

    }
}
```

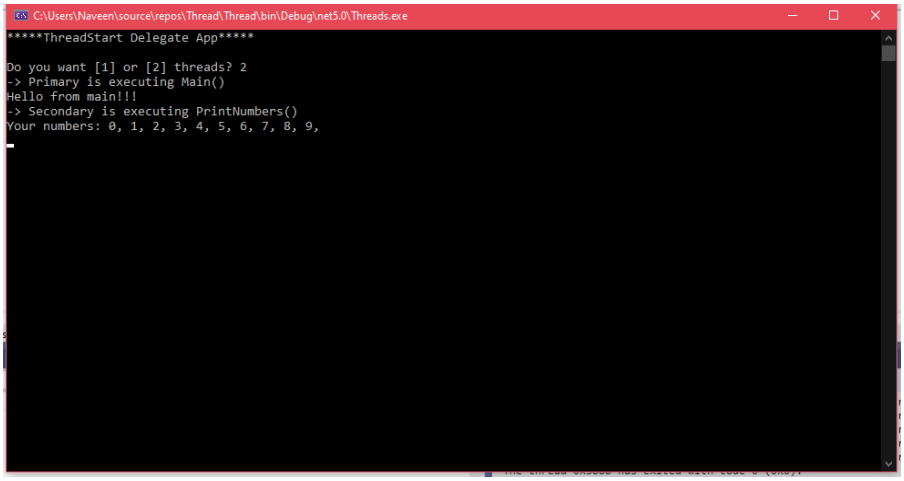
Program.cs

```
using System;
```

```
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading;
using System.Threading.Tasks;

namespace ThreadStartSample
{

    class Program
    {

        static void Main(string[] args)
        {
            Console.WriteLine("*****ThreadStart Delegate App*****\n");
            Console.Write("Do you want [1] or [2] threads? ");

            string threadCount = Console.ReadLine();

            // Name the current thread.
            Thread primaryThread = Thread.CurrentThread;
            primaryThread.Name = "Primary";

            // Display Thread info.
            Console.WriteLine("-> {0} is executing Main()", Thread.CurrentThread.Name);

            // Make worker class.
            Printer p = new Printer();

            switch (threadCount)
            {
                case "2":
                    // Now make the thread.
                    Thread backgroundThread = new Thread(new
ThreadStart(p.PrintNumbers));
                    backgroundThread.Name = "Secondary";
                    backgroundThread.Start();// Changes the state of current instance to
ThreadState.Running.
                    break;
                case "1":
                    p.PrintNumbers();
                    break;
                default:
                    Console.WriteLine("I don't know what you want... you get 1 thread.");
                    goto case "1";
            }

            Console.WriteLine("Hello from main!!!");
            Console.Read();

        }

    }

}
```

OUTPUT:





Multithreading - ThreadStart (MultithreadingSample)

Printer.cs

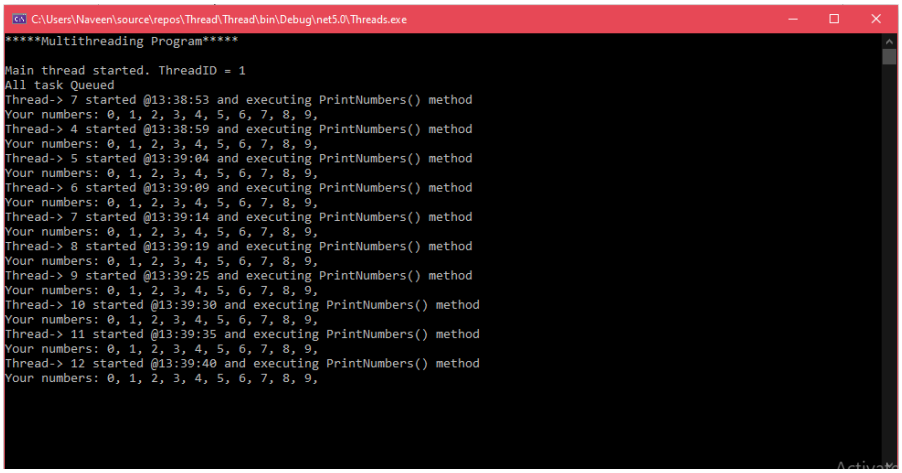
```
using System;
using System.Threading;

namespace Threads
{
    class Printer
    {
        private object lockToken = new object();
        public void PrintNumbers()
        {
            lock (lockToken)
            {
                // Display Thread info.
                Console.WriteLine("Thread-> {0} started @{1} and executing PrintNumbers()
method",
```

```
                    Thread.CurrentThread.ManagedThreadId,
DateTime.Now.ToLongTimeString());

                // Print out numbers.
                Console.Write("Your numbers: ");
                for (int i = 0; i < 10; i++)
                {
                    Console.Write("{0}, ", i);
                    Thread.Sleep(500);
                }
                Console.WriteLine();
            }
        }
    }
}
```

Program.cs

```
using System;
using System.Threading;

namespace Threads
{
    class Program
    {
        static void PrintTheNumbers(object state)
        {
            Printer task = (Printer)state;
            task.PrintNumbers();
        }
        static void Main(string[] args)
        {
            Console.WriteLine("*****Multithreading Program*****\n");

            Console.WriteLine("Main thread started. ThreadID = {0}",
                Thread.CurrentThread.ManagedThreadId);

            Printer p = new Printer();

            WaitCallback workItem = new WaitCallback(PrintTheNumbers);

            // Queue the method 10 times.
            for (int i = 0; i < 10; i++)
            {
                ThreadPool.QueueUserWorkItem(workItem, p);
            }
            Console.WriteLine("All task Queued");
            Console.ReadLine();
        }
    }

}
```

OUTPUT:


C:\Users\Naveen\source\repos\Thread\Thread\bin\Debug\net5.0\Threads.exe
*****Multithreading Program*****
Main thread started. ThreadID = 1
All task Queued
Thread-> 7 started @13:38:53 and executing PrintNumbers() method
Your numbers: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
Thread-> 4 started @13:38:59 and executing PrintNumbers() method
Your numbers: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
Thread-> 5 started @13:39:04 and executing PrintNumbers() method
Your numbers: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
Thread-> 6 started @13:39:09 and executing PrintNumbers() method
Your numbers: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
Thread-> 7 started @13:39:14 and executing PrintNumbers() method
Your numbers: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
Thread-> 8 started @13:39:19 and executing PrintNumbers() method
Your numbers: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
Thread-> 9 started @13:39:25 and executing PrintNumbers() method
Your numbers: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
Thread-> 10 started @13:39:30 and executing PrintNumbers() method
Your numbers: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
Thread-> 11 started @13:39:35 and executing PrintNumbers() method
Your numbers: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
Thread-> 12 started @13:39:40 and executing PrintNumbers() method
Your numbers: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9,