**Boppudi Vivek(916243)**

## C# Additional Topics Async Programming, Multithreading - Day 3 Handson 2.

**1. Multithreading - ThreadStart(Hands-On Lab)**

**print.cs**

**using System;**

**using System.Collections.Generic;**

**using System.Linq;**

**using System.Text;**

**using System.Threading.Tasks;**

**using System.Threading;**

**namespace ConsoleApp4**

**{**

**public class print**

**{**

**public void PrintNumbers()**

**{**

**// Display Thread info.**

**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.Tasks;**

**using System.Threading;**

**using ConsoleApp4;**

**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();**

**Thread primaryThread = Thread.CurrentThread;**

**primaryThread.Name = "Primary";**

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

**print p = new print();**

**switch (threadCount)**

**{**

**case "2":**

**Thread backgroundThread = new Thread(new ThreadStart(p.PrintNumbers));**

**backgroundThread.Name = "Secondary";**

**backgroundThread.Start();**

**break;**

**case "1":**

**p.PrintNumbers();**

**break;**

**default:**

**Console.WriteLine("I don't know what you want... you get 1 thread.");**

**goto case "1";**

**}**

**// Do some additional work.**

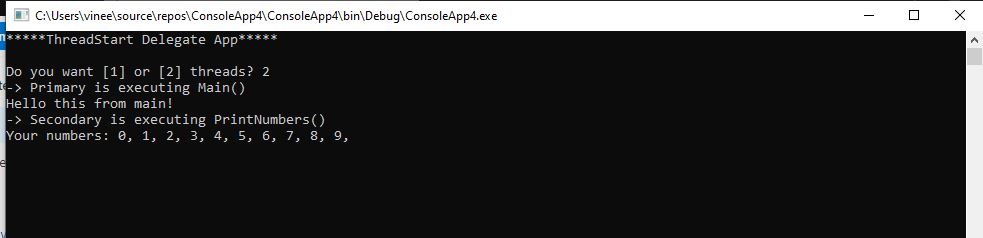
**Console.WriteLine("Hello this from main!");**

**Console.Read();**

**}**

**}**

**OUTPUT:**

****

**Multithreading – ThreadStart (MultithreadingSample)**

**print.cs**

**using System;**

**using System.Collections.Generic;**

**using System.Linq;**

**using System.Text;**

**using System.Threading.Tasks;**

**using System.Threading;**

**namespace ConsoleApp6**

**{**

**class Print**

**{**

**private object lockToken = new object();**

**public void PrintNumbers()**

**{**

**lock (lockToken)**

**{**

**Console.WriteLine("Thread-> {0} started @{1} and executing PrintNumbers() method",**

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

**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.Collections.Generic;**

**using System.Linq;**

**using System.Text;**

**using System.Threading.Tasks;**

**using System.Threading;**

**namespace ConsoleApp6**

**{**

**class Program**

**{**

**static void PrintTheNumbers(object state)**

**{**

**Print task = (Print)state;**

**task.PrintNumbers();**

**}**

**static void Main(string[] args)**

**{**

**Console.WriteLine("\*\*\*\*\*Multithreading Program\*\*\*\*\*\n");**

**Console.WriteLine("Main thread started. ThreadID = {0}",**

**Thread.CurrentThread.ManagedThreadId);**

**Print p = new Print();**

**WaitCallback workItem = new WaitCallback(PrintTheNumbers);**

**for (int i = 0; i < 10; i++)**

**{**

**ThreadPool.QueueUserWorkItem(workItem, p);**

**}**

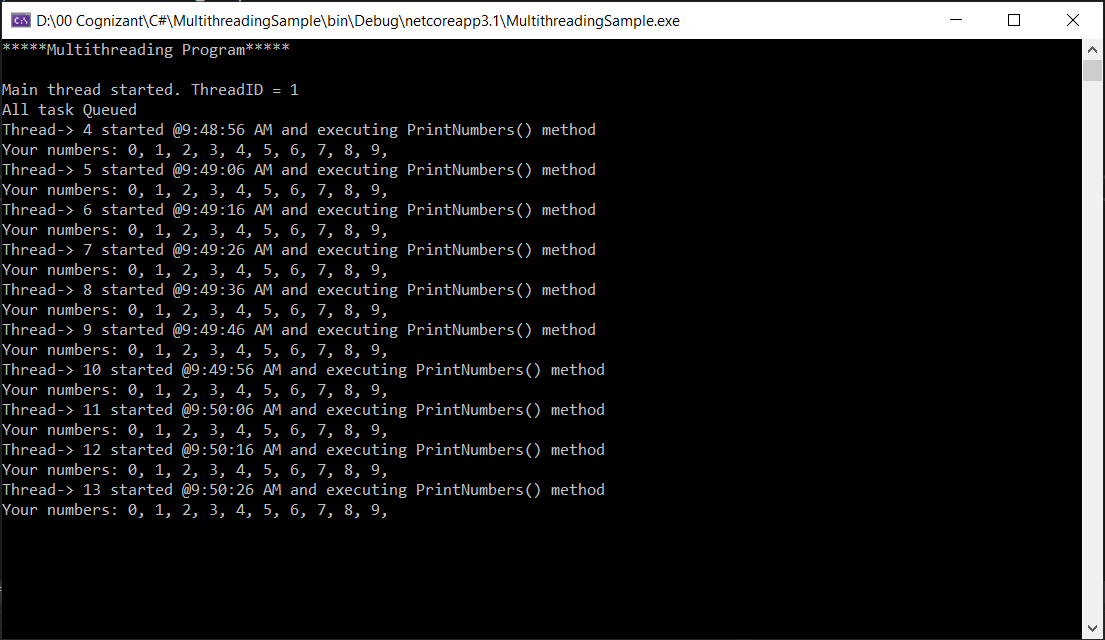
**Console.WriteLine("All task Queued");**

**Console.ReadLine();**

**}**

**}**

**}**

****