**S20: Multithreading**

**Multithreading: -** multiple threads work simultaneously. Process of achieve **multitasking**: - at the same time multiple task can be perform. For multithreading we use System. Threading namespace. It contain **process: -** it represent application where thread represent module of application. Its heavyweight component where thread is lightweight component. Whenever process has been created some memory will assign it automatically.

App1

App2

App3

Process

**Execution**

Inside process application will be run but inside thread application will execute. And in one application only one thread is present. By default when u create application thread will create. And called as main thread it works with single thread module (and its anonymous). In single thread if the existing method not won’t run completely till that no method will be execute further.

Thread executed using OS with **time-sharing**:- switching the method between each one each switch the thread... And it can be executed simultaneously. Once thread is declare and initialise then we need to start the thread then u will see the output. It will change the time as well. Each thread define unique flow control. Compilation code is there in app then need to set time according to that. Thread is used to create Child Thread.

EG:- Ms Word

S20\_\_MultiThreading.cs

using System;

using System.Collections.Generic;

using System.Reflection;

using System.Text;

using System.Threading;

namespace OOPS\_\_AllSession

{

class S20\_\_MultiThreading

{

public void PrintNumberList()

{

int i = 1;

do

{

Console.Write(i+", \t");

i++;

if (i == 30)

{

Console.WriteLine("\n\t\t----Going in Thread for Some Time----");

Thread.Sleep(9000);

}

} while (i <= 40);

}

public void PrintMultiplication()

{

int number = 2, i = 1;

do

{

Console.WriteLine($"{number} X {i} = " + number \* i);

i++;

} while (i <= 10);

}

public void PrintNamesArray()

{

string[] name = { "Amit", "Navyatha", "Eqbal", "Sayali", "\t\t", "Prakash", "Imran", "Abhilasha", "Aniket", "Arnav", "Ganesh" };

Console.Write("Names Are: ");

for (int i = 0; i < name.Length; i++)

{

if (i == 4)

{

Console.WriteLine("\n\t\t----Going in Thread for Some Time----");

Thread.Sleep(8000);

}

Console.Write(name[i] + " ");

}

Console.WriteLine();

}

}

}

OopsSessions.cs

using OOPS\_\_AllSession;

using System;

using System.Threading;

using static OOPS\_\_AllSession.S11\_\_ClassAndTypes;

namespace Oops\_\_AllSession

{

class OopsSessions

{

static void Main(string[] args)

{

Console.WriteLine("\*\*\*\*\*\*\*\*Welcome To Main Method\*\*\*\*\*\*\*\*\*\*\*");

//S20\_\_MultiThreading Threading

//Default Thread\_\_\_

Thread thread = Thread.CurrentThread;

thread.Name = "---Main Thread and Single.----";

Console.WriteLine("Printing Default Thread----");

//

S20\_\_MultiThreading multithreading = new S20\_\_MultiThreading();

//Time Sharing

//need to remove thread time from the condiion for runnig time sharing

Thread thread1 = new Thread(multithreading.PrintNamesArray);

Thread thread2 = new Thread(multithreading.PrintMultiplication);

Thread thread3 = new Thread(multithreading.PrintNumberList);

thread1.Start();

thread2.Start();

thread3.Start();

//multithreading.PrintNamesArray();

//multithreading.PrintMultiplication();

//multithreading.PrintNumberList();

}

}

}