**S21: AsyncAwait**

Asynchronous programming is very helpful. Using this Application can continue with the other work that does not depend on the completion of the entire task. Using async await we can take benefit of methods. Async await is depend on Task base Asynchronous Pattern (TAP). Where u can perform long task in application. It contains task in the form of object. Async await is used to perform api task along with ui. Async and await also known as code markers.

**Task:-**it represent operation or process that doesn’t return value and its executes asynchronously. Task object is one of the best central component

for task-based asynchronous pattern. It contain Iscanceled, IsCompleted, IsFaulted Status in it. And it’s presented in System.Threading.Task Namespace module.

**Async: -** it runs synchronously until it reaches first await expression at which point the method is suspended until awaited task is completed. If the method don’t contain await keyword it run asynchronously.

**Await:-**it suspend the evaluation of the enclosing async method until the asynchronous operation represented its operand completes. When the asynchronous operation complete it await method return the result. And without suspension of program it return the result.

S21\_\_AsyncAwait.cs

using System;

using System.Collections.Generic;

using System.Reflection;

using System.Text;

using System.Threading;

using System.Threading.Tasks;

namespace OOPS\_\_AllSession

{

class S21\_\_AsyncAwait

{

public int SyncMethod()

{

int result = 0;

for (int i = 1; i < 20; i++)

{

Thread.Sleep(400);

Console.WriteLine($"Sync Numbers are : {i}");

result += 1;

}

return result;

}

public Task<int> AsyncMethod()

{

Task<int> asyncMethod = new Task<int>(() =>

{

int result = 0;

for (int i = 1; i < 20; i++)

{

Thread.Sleep(400);

Console.WriteLine($"ASync Numbers are : {i}");

result += 1;

}

return result;

});

asyncMethod.Start();

return asyncMethod;

}

}

}

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\*\*\*\*\*\*\*\*\*\*\*");

S21\_\_AsyncAwait asyncAwait = new S21\_\_AsyncAwait();

Console.WriteLine("Calling ASync Method: ");

var asyncMethod = asyncAwait.AsyncMethod();

Console.WriteLine("Calling Sync Method: ");

var syncMethod = asyncAwait.SyncMethod();

}

}

}