using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Async16\_3

{

class Program

{

static void Main(string[] args)

{

GreetAsync().Wait();

}

public async static Task GreetAsync()

{

Console.WriteLine("Async Application Started");

// Call Async method

Task<string> greetMsg = GetGreetingsAsync();

// Do your other stuffs synchronously while async method is in progress

// The current thread that calls the GetGreetingdAsync() returns without waiting and

// continues to execute subsequent lines of code after this method call synchronouusly

// prints "Async Method in started....'while the async method GetGreetingsAsync() is still executing

// prints "Async method in started ..."while the async method GetGreetingsAsync() is still executing

Console.WriteLine("Async Method in started...");

Console.WriteLine();

Console.WriteLine("Cyrrent Time :" + DateTime.Now);

Console.WriteLine();

Console.WriteLine("Awaiting result from Async method..");

// All work completed ,wait for async method to complete

// When the thread subsequent executes the next line which has await operator (awit greeting),

//the thread suspends its execution and it is returned to the method that called GreetAsync()

// until the called async method(GetGreetingsAsync()) completes it execution and returns

string msg = await greetMsg;

// Print Async Method Result

Console.WriteLine("Async method completed!");

Console.WriteLine();

Console.WriteLine("Current Time : " + DateTime.Now);

Console.WriteLine();

Console.WriteLine("Async method output:" + greetMsg.Result);

Console.WriteLine("Async Application Ended!");

Console.Read();

}

public async static Task<string> GetGreetingsAsync()

{

// simulate long running process

await Task.Delay(10000);

return "Welcome to the NBA PlayOffs . Who's going to win?!";

}

}

}