Question Answering Bonus

1. ## What is Asynchronous programming ?

   Asynchronous programming is a technique that allows a program to execute tasks concurrently without blocking the main execution thread. This is particularly useful for operations that can take a long time to complete, such as file I/O, network requests, or database queries. Instead of waiting for such tasks to finish before moving on, asynchronous programming lets the program continue running other code and only resume the task when the awaited operation is complete.

   In C#, asynchronous programming is commonly implemented using the async and await keywords. When a method is marked with async, it can contain await expressions that pause the method's execution temporarily, freeing the main thread to do other work. Once the awaited task completes, execution resumes.

   Benefits of asynchronous programming include improved application responsiveness (especially in UI applications), better utilization of system resources, and the ability to handle multiple operations efficiently without blocking.

   For example, downloading data from a web service can be done asynchronously so the UI thread remains responsive while the download completes in the background.

   Example:

```
using System;
using System.Threading.Tasks;

class Program
{
```

```
    // The Main method marked async to allow awaiting async
calls inside
    static async Task Main()
    {
        Console.WriteLine("Starting async example...");

        // Call the asynchronous method and wait for it to
complete without blocking Main thread
        await LongRunningOperationAsync();

        Console.WriteLine("Async operation completed.");
    }

    // An example async method that simulates a long-running
operation
    static async Task LongRunningOperationAsync()
    {
        Console.WriteLine("Long running operation started...");

        // Await asynchronously for 3 seconds without blocking
        await Task.Delay(3000);

        Console.WriteLine("Long running operation finished.");
    }
}
```

- async Task Main() lets the program entry point be asynchronous.
- LongRunningOperation*A*sync() is an asynchronous method simulating work using Task.Delay.
- await Task.Delay(3000) pauses this method asynchronously for 3 seconds, freeing the main thread during this wait.

- The program prints messages before, during, and after the async operation to show the flow.