Senior Developer Technical Assessment – Debugging & Code Review

Introduction

As part of our selection process for **Senior Developer (Java / C#)**, we want to evaluate your ability to **analyze existing code**, identify problems, and propose improvements.

In this test, you will be given two code snippets (one in **Java**, one in **C#**). Each snippet contains intentional issues related to **thread safety, resource management, asynchronous execution, and clean coding practices**.

Instructions

1. Review the provided code carefully.
2. Identify all problems you see (bugs, bad practices, scalability issues, security risks, etc.).
3. For each problem, explain **why it is an issue** and **how you would fix it**.
4. You may write your answers in text, or include corrected code snippets where relevant.
5. There is **no single correct answer** – we are evaluating your reasoning, experience, and ability to justify technical decisions.

Part 1 – Java Snippet

```
public class FileProcessor {
    private static List<String> lines = new ArrayList<>();

    public static void main(String[] args) throws Exception {
        ExecutorService executor = Executors.newFixedThreadPool(5);

        for (int i = 0; i < 10; i++) {
            executor.submit(() -> {
                try {
                    BufferedReader br = new BufferedReader(new
FileReader("data.txt"));
                    String line;
                    ~~while ((line = br.readLine()) != null) {~~
                        ~~lines.add(line.toUpperCase());~~
                    ~~}~~
                    br.close();
                } catch (Exception e) {
                    e.printStackTrace();
                }
            });
        }
        executor.shutdown();
        System.out.println("Lines processed: " + lines.size());
    }
}
```

Part 2 – C# Snippet

```
public class Downloader
{
    private static List<string> cache = new List<string>();

    public static async Task Main(string[] args)
    {
        for (int i = 0; i < 10; i++)
        {
            DownloadAsync("https://example.com/data/" + i);
        }

        Console.WriteLine("Downloads started");
        Console.WriteLine("Cache size: " + cache.Count);
    }

    private static async Task DownloadAsync(string url)
    {
        using (HttpClient client = new HttpClient())
        {
            var data = await client.GetStringAsync(url);
            cache.Add(data);
        }
    }
}
```

Your Task

For each snippet:

1. Identify the problems (bugs, concurrency issues, inefficiencies, bad practices).
2. Explain **why** each problem is critical in a real-world system.

3. Suggest improvements (or provide a corrected version of the code if you prefer).

✅ **Deliverable:** Please return an executable code, stored in you personal git repository, and a Readme.md file with your analysis.