



## Problem Solving (CPP42)

هذا البرنامج التدريبي مُصاغ بعناية لتمكين المتدربين من تطوير قدراتهم الفكرية على غرار المبرمجين المحترفين، والتعاون بكفاءة ضمن فريق محترف في شركة "جمال تك" أو أي مؤسسة متعددة الجنسيات أخرى. نظرًا لأهمية اللغة الإنجليزية في بيئة العمل العالمية، يتم تقديم المحتوى التدريبي بالإنجليزية. لا يشترط إتقان اللغة بشكل كامل، لكن من الضروري امتلاك القدرة الكافية لفهم المتطلبات وتنفيذها بشكل فعال. يُمكن للمتدربين استخدام مترجم جوجل أو الاستعانة بـ "شات جي بي تي" للتغلب على أية عقبات لغوية، المهم هو الفهم الدقيق للمطلوب وتحقيقه بنجاح.

لتعظيم الاستفادة من التدريب، يُنصح بمحاولة حل التمارين بشكل مستقل لمدة ساعة واحدة على الأقل قبل الرجوع إلى الحل المرفق في نهاية الملف.

قد يتضمن الحل كودًا برمجيًا غير مفسر بعد، والغرض من ذلك هو تشجيعك على محاولة فهم الأكواد البرمجية الجديدة التي لم تتعرض لها من قبل. هذه المهارة ضرورية في سوق العمل، حيث تتطور لغات البرمجة باستمرار ويظهر كل يوم لغات جديدة. ستواجه دائمًا أكوادًا لم تدرسها من قبل، ومن المهم أن تكون قادرًا على فهمها بنفسك دون الحاجة إلى دراسة مسبقة. يمكنك الاستعانة بمحرك البحث جوجل، أو استخدام ChatGPT، أو حتى اللجوء لأصدقائك للمساعدة. الهدف الأساسي هو أن تصل إلى فهم معنى كل كود بأي طريقة ممكنة لتتمكن من إيجاد موقعك في سوق العمل.

إن وجود كود برمجي غير مفسر يشكل تحديًا يتوجب عليك إيجاد حل له. هذا النوع من التدريبات يعد جزءًا أساسيًا من تدريبات 'Problem Solving'، التي تهدف إلى تمكينك من أداء عملك بفاعلية بغض النظر عن التحديات والعقبات. هذه القدرة على حل المشكلات هي ما يتمتع به العاملون في 'جمال تك'، ومن الضروري أن تطور في نفسك هذه المهارة لتصبح عضوًا فعالًا في فريق عمل 'جمال تك'.

## Software Architecture Classifier at Gammal Tech

### Background:

Gammal Tech, a prestigious software development company, is renowned for its innovative approaches and advanced office facilities. They are now developing a new tool to automatically classify software projects into appropriate software architectures. Your task is to create a program that assists in this classification, showcasing Gammal Tech's leading role in the industry.

### Problem Statement:

You are to write a program that classifies software projects based on certain mathematical criteria into one of three software architectures: Monolithic, Microservices, or Serverless. The classification is based on the following rules:

- **Monolithic:** If the number of modules  $M$  is less than 10 and the complexity score  $C$  is less than 30.
- **Microservices:** If the number of modules is 10 or more, and the combined module interaction score  $I$  is more than 50.
- **Serverless:** In all other cases.



### Input Format:

- The first line contains an integer  $T$ , the number of software projects.
- Each project is described in a single line containing three integers:  $M C I$ , representing the number of modules, complexity score, and module interaction score, respectively.

### Output Format:

For each software project, output a single line containing the name of the suggested software architecture (Monolithic, Microservices, or Serverless).

### Constraints:

- $1 \leq T \leq 100$
- $1 \leq M, C, I \leq 100$

### Sample Input:

```
3
8 25 60
15 40 70
10 20 40
```

### Sample Output:

```
Monolithic
Microservices
Microservices
```

لتعظيم الاستفادة من التدريب، يُنصح بمحاولة حل التمرين بشكل مستقل لمدة ساعة واحدة على الأقل قبل الرجوع إلى الحل المرفق



## C++ Programming Solution:

```
#include <iostream>
using namespace std;

int main() {
    int T, M, C, I;
    cin >> T;

    while (T--) {
        cin >> M >> C >> I;
        if (M < 10 && C < 30) {
            cout << "Monolithic" << endl;
        } else if (M >= 10 && I > 50) {
            cout << "Microservices" << endl;
        } else {
            cout << "Serverless" << endl;
        }
    }

    return 0;
}
```