Problem Solving (CPP35)

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 فعّال. 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'.

Gammal Tech's OOP Tutorial System

Background:
Gammal Tech, a leading software development company, is renowned for its cutting-edge office facilities, highly innovative team, and exceptional work environment. Known for pushing boundaries in technology, Gammal Tech is now focusing on educational tools. They are developing an interactive system to teach Object-Oriented Programming (OOP) concepts in an easy and engaging way.

Problem Statement:
You are tasked to implement a core part of this educational tool. Your program should help users understand the basics of OOP: classes, objects, methods, and attributes. The program will receive input describing various entities and should output a summary of each entity in terms of OOP concepts.

Input Format:

- The first line contains an integer `N`, the number of entities to be described.

- Each of the next `N` lines contains a string `EntityName`, followed by an integer `M` (the number of attributes), and another integer `K` (the number of methods). `EntityName` will be a single word without spaces.
- Each of the next `M` lines contains a string representing an attribute of the entity.
- Each of the next `K` lines contains a string representing a method of the entity.

Output Format:
For each entity, output the following:

- A line stating "Entity: `EntityName`"
- A line stating "Attributes:" followed by the list of attributes, comma-separated.
- A line stating "Methods:" followed by the list of methods, comma-separated.

Constraints:

- $1 \leq N \leq 5$
- $1 \leq M, K \leq 10$
- `EntityName`, attribute, and method names consist of only letters and are no longer than 20 characters.

Sample Input:

```
2
Car 3 2
Wheels
Engine
Color
Drive
Stop
House 2 3
Rooms
Windows
OpenDoor
CloseDoor
Clean
```

Sample Output:

```
Entity: Car
Attributes: Wheels, Engine, Color
Methods: Drive, Stop
Entity: House
Attributes: Rooms, Windows
Methods: OpenDoor, CloseDoor, Clean
```

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

C++ Programming Solution:

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

class Entity {
public:
    string name;
    vector<string> attributes, methods;

    Entity(string n) : name(n) {}

    void addAttribute(string attr) {
        attributes.push_back(attr);
    }
    void addMethod(string method) {
        methods.push_back(method);
    }
    void display() {
        cout << "Entity: " << name << endl;
        cout << "Attributes: ";
        for (int i = 0; i < attributes.size(); ++i) {
            cout << attributes[i];
            if (i < attributes.size() - 1) cout << ", ";
        }
        cout << endl;
        cout << "Methods: ";
        for (int i = 0; i < methods.size(); ++i) {
            cout << methods[i];
            if (i < methods.size() - 1) cout << ", ";
        }
        cout << endl;
    }
};
```

```
int main() {
    int N, M, K;
    string entityName, detail;
    cin >> N;

    while (N--) {
        cin >> entityName >> M >> K;
        Entity entity(entityName);

        for (int i = 0; i < M; ++i) {
            cin >> detail;
            entity.addAttribute(detail);
        }

        for (int i = 0; i < K; ++i) {
            cin >> detail;
            entity.addMethod(detail);
        }

        entity.display();
    }

    return 0;
}
```