

دليل التمارين التدريجية (30 - Gym Management System)

تمرين

مقدمة

هذا الدليل يحتوي على 30 تمرين تدريجي لبناء نظام إدارة النادي الرياضي خطوة بخطوة. كل تمرين يبني على السابق، من الأساسية البسيطة إلى المشروع الكامل.

القاعدة الذهبية:

- لا تنتقل للتمرين التالي إلا بعد إتمام الحالي بنجاح
- احفظ كل تمرين في ملف منفصل (ex1.cpp, ex2.cpp, ...)
- اختبر كل تمرين جيداً قبل الانتقال

خارطة الطريق



المراحل 1 (تمارين 1-6) → الأساسية والدوال المساعدة

المراحل 2 (تمارين 7-12) → إدارة الأعضاء الأساسية 

المراحل 3 (تمارين 13-18) → الاشتراكات والتاريخ

المراحل 4 (تمارين 19-24) → الحضور والمدفوعات

المراحل 5 (تمارين 25-30) → التقارير والمشروع الكامل 



المراحل الأولى: الأساسية

1 تمرين : أول Structure

الهدف 

تعلم الأساسية Structures.

المطلوب:



cpp

```
struct Member {  
    string memberID;  
    string name;  
    int age;  
};
```

- أنشئ عضو واحد
- اطلب البيانات من المستخدم
- اطبعها

المطلوب:



```
Enter ID: GYM001  
Enter Name: Ahmad  
Enter Age: 25
```

```
Member Info:  
ID: GYM001  
Name: Ahmad  
Age: 25
```

تمرين ②: دالة طباعة منسقة

الهدف:

Functions مع Structures.

المطلوب:



cpp

```
void PrintMemberCard(Member m);
```

- طبع 3 أعضاء بشكل منسق

 **Output:**



```
|| MEMBER CARD ||  
|| ID: GYM001 ||  
|| Name: Ahmad ||  
|| Age: 25 ||
```

تمرين 3: حفظ في ملف (Write)

 **الهدف:**

الكتابة - File Handling.

 **المطلوب:**

- احفظ عضو واحد في member.txt
- الصيغة: ID#/##Name##/#Age

 **محتوى الملف:**



GYM001##Ahmad##25

تمرين 4: قراءة من ملف (Read)

 **الهدف:**

القراءة - File Handling.

المطلوب:

- اقرأ العضو من الملف
- اطبع بياناتة

5 : دالة **SplitString** تمرين

الهدف:

تقسيم النصوص.

المطلوب:



cpp

```
vector<string> SplitString(string text, string delim);
```

مثال:



Input: "A##B##C"

Output: ["A", "B", "C"]

كود مساعد:



cpp

```
vector<string> SplitString(string text, string delim) {
    vector<string> result;
    size_t pos;
    while ((pos = text.find(delim)) != string::npos) {
        result.push_back(text.substr(0, pos));
        text.erase(0, pos + delim.length());
    }
    if (text != "") result.push_back(text);
    return result;
}
```

تمرين 6: دوال التحويل

الهدف:

تحويل Line ↔ Struct.

المطلوب:



cpp

```
string ConvertToString(Member m);
Member ConvertToString(string line);
```

اختبار:



Member → Line → Member

تأكد أن البيانات ما تتغير



المراحل الثانية: إدارة الأعضاء

7 كاملة Structure : تمرين

الهدف :

توسيع الـ Structure.

المطلوب :



cpp

```
struct Member {  
    string memberID;  
    string fullName;  
    string phone;  
    string email;  
    int age;  
    string gender;  
    string joinDate;  
    bool isActive;  
};
```

- حدّث جميع الدوال السابقة

8 تمرين : حفظ عدة أعضاء

الهدف :

إلى ملف

المطلوب :

- استخدم ios::app
- احفظ 3 أعضاء
- كل واحد في سطر

نصيحة :



cpp

```
file.open("members.txt", ios::app); // append!
```

تمرين 9: قراءة جميع الأعضاء

الهدف:

Vector من الملف.

المطلوب:



cpp

```
vector<Member> LoadAllMembers();
```

كود:



cpp

```
vector<Member> LoadAllMembers() {
    vector<Member> members;
    fstream file("members.txt", ios::in);
    string line;
    while (getline(file, line)) {
        members.push_back(ConvertToMember(line));
    }
    return members;
}
```

تمرين 10: جدول الأعضاء

الهدف:

عرض احترافي.

المطلوب:



cpp

```
void ShowAllMembers();
```

Output:



MEMBERS (3)				
ID	Name	Phone	Age	Status
GYM001	Ahmad Hassan	0599123456	25	<input checked="" type="checkbox"/>
GYM002	Sara Ali	0598765432	30	<input checked="" type="checkbox"/>

استخدم:



cpp

```
#include <iomanip>
cout << setw(10) << left << data;
```

تمرين 1: البحث بال ID

الهدف:

Search في Vector.

المطلوب:



cpp

```
bool FindByID(string id, vector<Member> v, Member& result);
```

Output:



Enter ID: GYM001

 Found!

[عرض البطاقة]

1 2 : التتعديل (Update) تمرين

الهدف:

تحديث البيانات.

المطلوب:

1. ابحث عن عضو.
2. عدل بياناتة.
3. كله في الملف vector احفظ الـ .

استراتيجية:



1. Load all → vector
 2. Find & update in vector
 3. Save all back to file
-



إنجاز!

بعد التمرين 12: عندك نظام أعضاء كامل



المراحلة الثالثة: الاشتراكات

1 3 تمرين: Membership Structure

المطلوب:



cpp

```
struct Membership {  
    string membershipID;  
    string memberID;  
    string type;      // Monthly, Yearly  
    string startDate;  
    string endDate;  
    double price;  
    bool isActive;  
};
```

1 4

تمرين: حساب تاريخ الانتهاء

المطلوب:



cpp

```
int GetDuration(string type); // Monthly→30, Yearly→365  
string CalculateEndDate(string start, int days);
```



01/01/2025 + 30 days = 31/01/2025

تمرين 1 5: حفظ الاشتراكات

المطلوب:

- نفس المفهوم: Save/Load
- ملف: memberships.txt

تمرين 1 6: ربط العضو بالاشتراك

المطلوب:

- اعرض بطاقة عضو + اشتراكه

Output:



```
|| MEMBER: Ahmad      ||
|| MEMBERSHIP: Monthly ||
|| Days Left: 15      ||
```

تمرين 1 7: فحص الصلاحية

المطلوب:



cpp

```
bool IsValid(Membership m);  
int DaysRemaining(Membership m);
```

١٨ : تمرين التجديد

المطلوب:

- ابحث عن اشتراك منتهي
- أنشئ اشتراك جديد
- تاريخ البداية = تاريخ انتهاء القديم



المرحلة الرابعة: الحضور والمدفوعات

١٩ : تمرين Attendance Structure

المطلوب:



cpp

```
struct Attendance {  
    string attendanceID;  
    string memberID;  
    string date;  
    string checkInTime;  
    string checkOutTime;  
    int durationMinutes;  
};
```

٢٠ : تمرين Check-In

المطلوب:

- اطلب Member ID.
- تحقق من الاشتراك.
- سجل الدخول مع الوقت.

Output:



Enter ID: GYM001

 Valid membership!

Check-in: 08:30

Welcome Ahmad! 

2 1: تمرين Check-Out

المطلوب:

- ابحث عن آخر check-in
- سجل الخروج
- احسب المدة

2 2: تمرين سجل الحضور

المطلوب:

- اعرض جميع حضور عضو معين
- احسب: عدد الزيارات، متوسط المدة

2 3: تمرين Payment Structure

المطلوب:



cpp

```
struct Payment {  
    string paymentID;  
    string memberID;  
    double amount;  
    string date;  
    string method; // Cash, Card  
};
```

تمرين 24: تسجيل دفعات



- المطلوب:
- سجل دفعات عند شراء اشتراك
 - احفظ في payments.txt

المرحلة الخامسة: التقارير والانهاء

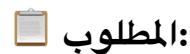
تمرين 25: إحصائيات الأعضاء



cpp

```
int CountActive();  
int CountInactive();  
double AverageAge();
```

تمرين 26: إحصائيات الاشتراكات



cpp

```
int CountExpired();  
int CountExpiringSoon() // < 7 days  
string MostPopularType();
```

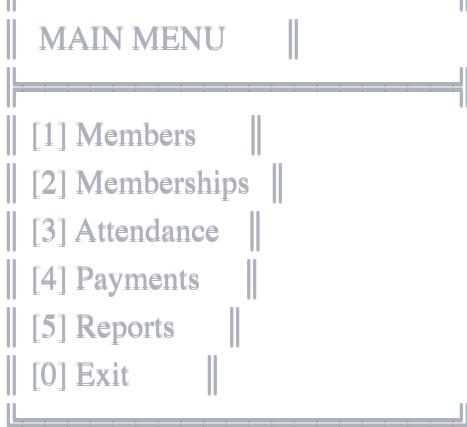
تمرين 27: التقرير العام



اعرض:

- عدد الأعضاء
 - عدد الاشتراكات
 - الحضور اليوم
 - الإيرادات الشهرية
-

تمرين 28: القوائم (Menus)



تمرين 29: دمج كل شيء



- جمع جميع الوظائف
- قوائم متداخلة

- navigation سليم

تمرين 30: المشروع النهائي

المطلوب:

1. تعلم جميع الـ Features
2. Error Handling
3. Input Validation
4. تنسيق احترافي
5. README.md
6. رفع على GitHub

نصائح النجاح

✓: افعل:

1. اختبر كل تمرين جيداً.
2. اكتب تعليقات.
3. استخدم أسماء واضحة.
4. حفظ نسخ احتياطية.

✗: لا تفعل:

1. لا تنتقل قبل الإتمام.
2. لا تنسخ بدون فهم.
3. لا تهمل الاختبار.
4. لا تكتب كل شيء دفعة واحدة.

جدول التقدم

التمرين	ال موضوع	الحالات
1-6	أساسيات	█
7-12	أعضاء	█
13-18	اشتراكات	█
19-24	حضور و مذفو عات	█
25-30	تقارير و إنتهاء	█



النتيجة النهائية

بعد إتمام الـ 30 تمرین:

- مشروع كامل واحترافي
- فهم عميق لـ C++
- خبرة في File Handling
- مهارات حل المشاكل
- قوي Portfolio

🎉 بال توفيق! ابدأ من التمرین 1 الآن