

## الدرس السابع : السجلات

### Structures



#### 1- تعريف السجلات

السجل عبارة عن مجموعة مترابطة من البيانات المختلفة التي تسمى حقولاً (fields) أو أعضاء (members). في لغة C++، يتم تعريف السجلات بإتباع قاعدة البناء التالية:

```
struct Structure_Name
{
    Data_Type1  Field1;
    Data_Type2  Field2;
    ...
};
```

حيث Structure\_Name هو اسم السجل ، ويدخل السجل تتوالى الحقول المختلفة (field1، field2، ...) ولكل حقل نوعه الخاص.

بعد تعريف السجل يمكننا تعريف متغيرات من نوع هذا السجل لاستخدامها في البرنامج حسب الحاجة وذلك بإتباع الصيغة التالية:

```
Structure_Name Variable_Name;
```

#### مثال

```
struct Person
{
    char  Name[20];
    int   Age;
};
Person  p1, p2, p3;
```

p1، p2 و p3 متغيرات من فئة Person (أي أشخاص).

#### ملاحظة

الفرق الرئيسي بين المصفوفة والسجل هو أن بيانات المصفوفة تنتمي إلى نفس النوع بينما تنتمي بيانات السجل إلى أنواع مختلفة.

## 2- معالجة حقول السجل

يتم الوصول إلى حقول السجل باستخدام مؤثر النقطة (.) بين اسم المتغير من فئة السجل واسم الحقل أي:

Variable\_Name.Field

### مثال

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

int main()
{
    struct Person
    {
        char Name[20];
        int Age;
    };
    Person p1, p2, p3;
    strcpy(p1.Name, "Ahmed");
    p1.Age = 20;
    strcpy(p2.Name, "Fahd");
    p2.Age = p1.Age + 1;
    p3 = p1; // إسناد متغير من فئة السجل إلى متغير آخر من نفس النوع
    cout<<"First person:"<<"\t"<<p1.Name<<"\t"<<p1.Age<<" years"<<endl;
    cout<<"Second person:"<<"\t"<<p2.Name<<"\t"<<p2.Age<<" years"<<endl;
    cout<<"Third person:"<<"\t"<<p3.Name<<"\t"<<p3.Age<<" years"<<endl;
    return 0;
}
```

### مخرجات البرنامج

```
First person:  Ahmed  20 years
Second person:  Fahd   21 years
Third person:   Ahmed  20 years
Press any key to continue . . . _
```

### تمرين رقم 1

أكتب برنامجاً بلغة C++ يستخدم تقنية السجلات لتمثيل البيانات التي تخص الطلبة:

- الرقم الجامعي (StudID)
- الاسم (Sname)
- العنوان (Address)
- تاريخ الميلاد (DOB)

يحتوي تاريخ الميلاد على ثلاثة حقول:

- اليوم (Day)
- الشهر (Month)
- السنة (Year).

## • الحل

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

int main()
{
    struct date
    {
        int day;
        int month ;
        int year;
    };
    struct student
    {
        int StudID;
        char Sname[20];
        char Address[40];
        date DOB;
    };
    student stud;
    cout<<"Enter the student number : ";
    cin>>stud.StudID;
    cout<<"Enter the name : ";
    cin>>stud.Sname;
    cout<<"Enter the address : ";
    cin>>stud.Address;
    cout<<"Enter the date of birth (Day Month Year) : ";
    cin>>stud.DOB.day>>stud.DOB.month>>stud.DOB.year;
    cout<<endl;
    cout<<"Student No "<<stud.StudID<<endl;
    cout<<"\tName = "<<stud.Sname<<endl;
    cout<<"\tAddress = "<<stud.Address<<endl;
    cout<<"\tBirth Date = "<<stud.DOB.day<<"/"<<stud.DOB.month<<"/"<<stud.DOB.year;
    cout<<endl;
    return 0;
}
```

## • تتبع تنفيذ البرنامج

```
Enter the student number : 420320070
Enter the name : Bandar
Enter the address : Jeddah
Enter the date of birth (Day Month Year) : 20 10 1988

Student No 420320070
Name = Bandar
Address = Jeddah
Birth Date = 20/10/1988

Press any key to continue . . .
```

## تمرين رقم 2

قم بإصلاح الأخطاء الواردة في البرنامج التالي:

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

int main( )
{
    Struct car
    {
        char Mark[20];
        int Model;
        int Price;
    }

    car car1, car2, car3;

    car1.Mark = "Toyota Camry";
    car1.Model = 2007;
    car1.Price = 70000;

    strcpy(car2.Mark, "Toyota Corolla");
    car2.Model = 2006;
    car2.Price = 40000;

    strcpy(car3.Mark, "Toyota Echo");
    car3.Model = 2007;
    car3.price = 35000;

    cout<<"MARK"<<"\t\t"<<"MODEL"<<"\t"<<"PRICE"<<endl;
    cout<<"----"<<"\t\t"<<"----"<<"\t"<<"----"<<endl;
    cout<<car1.Mark<<"\t"<<car1.Model<<"\t"<<car1.Price<<endl;
    cout<<car2.Mark<<"\t"<<car2.Model<<"\t"<<car2.Price<<endl;
    cout<<car3.Mark<<"\t"<<car3.Model<<"\t"<<car3.Price<<endl;
    cout<<endl;
    return 0;
}
```

## المخرجات المنتظرة على الشاشة

MARK	MODEL	PRICE
Toyota Camry	2007	70000
Toyota Corolla	2006	40000
Toyota Echo	2007	35000

Press any key to continue . . . \_

### 3- تعيين قيم أولية للحقول initializing fields

تسمح لغة C++ بإسناد قيما أولية للحقول عند التصريح بمتغير من فئة السجل كما في المثال التالي:

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

int main( )
{
    struct product
    {
        int reference;
        char name[20];
        float price;
    };
    product p1 = {1001, "apple", 5};
    cout<<"Product p1"<<endl;
    cout<<"\treference = "<<p1.reference<<endl;
    cout<<"\tname = "<<p1.name<<endl;
    cout<<"\tprice = "<<p1.price<<endl;
    return 0;
}
```

مخرجات البرنامج

```
Product p1
      reference = 1001
      name = apple
      price = 5
Press any key to continue . . . _
```

### 4- مؤشرات السجلات

مثل بقية أنواع البيانات، يمكن استخدام مؤشرات نحو متغيرات من فئة السجل. يتم الوصول إلى حقول السجل عن طريق المؤشر باستخدام إحدى الصيغ التالية:

Pointer -> Field

أو

(\*Pointer).Field

مثال

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

int main( )
{
    struct product
    {
        int reference;
        char name[20];
        float price;
    };
}
```

```

product p1 = {1001, "apple", 5};
product *ptr = &p1;
strcpy(ptr -> name, "peach");
ptr -> price = 6.5;
cout<<"Product p1"<<endl;
cout<<"\treference = "<<p1.reference<<endl;
cout<<"\tname = "<<p1.name<<endl;
cout<<"\tprice = "<<p1.price<<endl;
return 0;
}

```

### مخرجات البرنامج

```

Product p1
reference = 1001
name = peach
price = 6.5

Press any key to continue

```

### 5- مصفوفات السجلات

لإنشاء مصفوفة من السجلات يجب إتباع المراحل التالية:

1- الإعلان عن السجل

2- الإعلان عن مصفوفة تكون عناصرها من فئة هذا السجل.

ل للوصول إلى عنصر معين من عناصر المصفوفة نستخدم الصيغة التالية:

```
Array_Name[Index]
```

ل للوصول إلى أحد حقول هذا العنصر نستخدم الصيغة التالية:

```
Array_Name[Index].Field
```

### مثال

```

#include <iostream>
#include <iomanip>
using namespace std;
int main( )
{
    struct product
    {
        int reference;
        char name[20];
        float price;
    };
    product P[3];
    int i;
    for (i = 0; i <= 2; i++)
    {
        cout<<"Enter the reference of the product : ";

```

```

        cin>>P[i].reference;
        cout<<"Enter the name : ";
        cin>>P[i].name;
        cout<<"Enter the Price : ";
        cin>>P[i].price;
        cout<<"-----" <<endl;
    }

    cout<<"REFERENCE\tNAME\tPRICE\n";
    cout<<"-----\t----\t-----\n";
    for (i = 0; i <= 2; i++)
    {
        cout<<setw(8)<<P[i].reference<<"\t"<<P[i].name<<"\t"<<P[i].price;
        cout<<endl;
    }
    cout<<endl;
    return 0;
}

```

### تتبع تنفيذ البرنامج

```

Enter the reference of the product : 1001
Enter the name : Orange
Enter the Price : 2.5
-----
Enter the reference of the product : 1002
Enter the name : Apple
Enter the Price : 4.5
-----
Enter the reference of the product : 1003
Enter the name : Bananas
Enter the Price : 3.25
-----
REFERENCE      NAME      PRICE
-----
1001           Orange   2.5
1002           Apple    4.5
1003           Bananas  3.25
Press any key to continue . . .

```

**ملاحظة:** يقوم الأمر `setw` بتحديد عرض الحقل أو المتغير الذي سيتم عرض قيمته على الشاشة. لاستخدام هذه الدالة لا بد من إدراج الملف الرئيسي `<iomanip>` في أول البرنامج.