

# جامعة المنارة

مقرر برمجة 2

د. هلا نصار

م. أوشين داود

جامعة  
المنارة

HANARA UNIVERSITY

محاضرات الأسبوع 4

الفصل الأول - 2022/2021

## Object –Oriented Programming (OOP) in C++

تعلمنا سابقاً البرمجة الإجرائية Procedural Programming تعتمد على كتابة الاجرائيات procedures والتوابع functions التي تقوم بتنفيذ العمليات على البيانات

اما البرمجة غرضية التوجه Object –Oriented Programming فهي تعتمد على تكوين كائنات objects تتضمن كلاً من البيانات والتوابع معاً.

تتفوق البرمجة كائنية التوجه على البرمجة الإجرائية بالنقاط التالية:

- تعتبر البرمجة غرضية التوجه هي الأسرع والاسهل في التنفيذ
- تؤمن البرمجة غرضية التوجه بنية واضحة للبرامج
- تجعل البرنامج خال من التكرارات واسهل في الصيانة والتعديل
- تتيح إمكانية خلق تطبيقات مختلفة من نفس البرنامج بتعليمات اقل ووقت اقصر للتطوير والتعديل.

## Struct in C++

السجل Struct في لغة C++ هو بنية معطيات معرفة من قبل المستخدم user defined data type تستخدم لتخزين أنواع مختلفة من المعطيات معاً.

بفرض نحتاج تخزين معلومات عن كل شخص : الاسم – رقم الهاتف – العنوان – العمر.  
نستطيع تخزين هذه البيانات عن كل شخص في بنية معطيات واحدة هي السجل Struct ويسمى كل عنصر من السجل حقل field في السجل. أما كل متحول من نوع هذا السجل struct هو حالة من السجل Instances of a structure

صيغة تعريف السجل: يبدأ بكلمة مفتاحية struct

```
struct struct_name
{
    // struct members
};
```

## Struct in C++: Basics

```
struct Person
{
    string name;
    int citizenship;
    int age;
}
```

```
Person p;
```

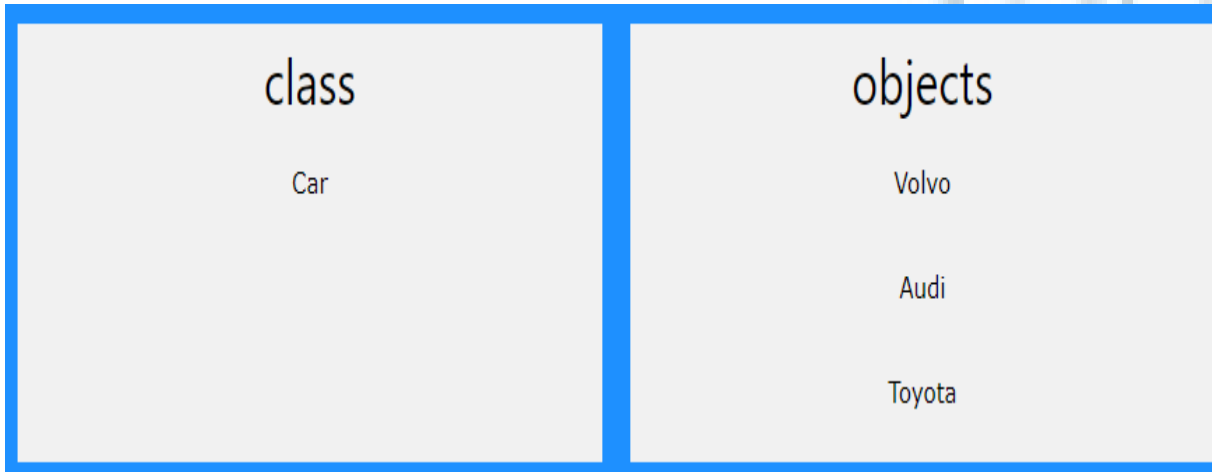
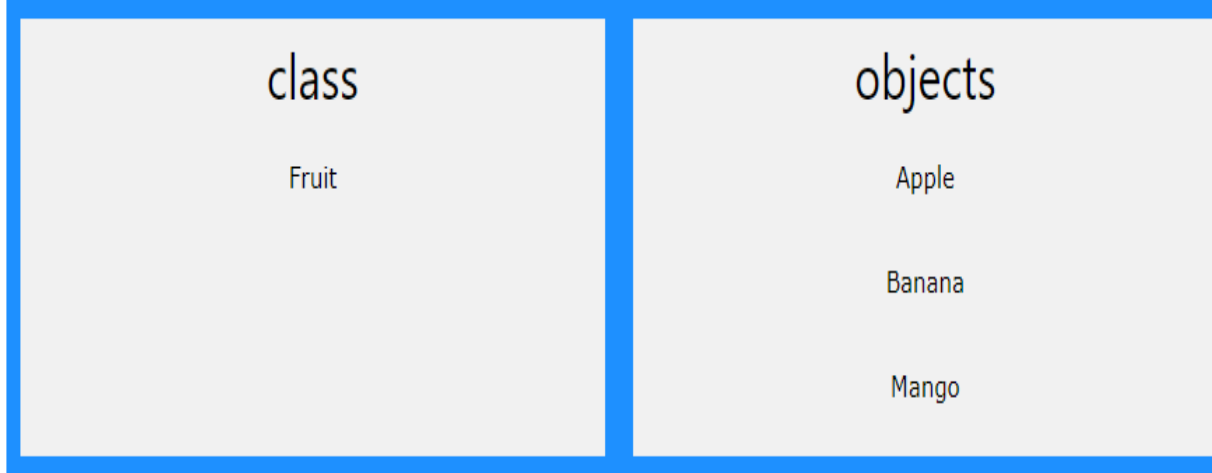
```
p.age = 27;
```

يتم تعريف سجل اسمه Person بثلاثة أعضاء: الاسم- الإقامة – العمر  
نلاحظ وجود أنواع معطيات مختلفة.  
يتم التصريح عنه ضمن قسم التضمينات قبل main وجدير ذكره أنه عند انتهاء  
هذا التصريح لا يكون قد تم تخصيص مكان للسجل في الذاكرة.

بهذا التصريح تم تعريف نوع معطيات جديد هو Person ومنه نستطيع ان نعرف  
متحولات مختلفة من نوع السجل Person عندها تخصص ذاكرة لكل سجل  
ويسمى حالة السجل Person (struct Instance).

للوصول الى احد حقول السجل p نستطيع استخدام معامل "." مع اسم الحقل.

## Classes & Objects in C++



تم تطوير السجلات للحصول على الأصناف حيث تعتبر الأصناف Classes والكائنات Objects البنى الأساسية في البرمجة كائنية التوجه وتبين الأمثلة المجاورة علاقة الصنف والكائن.

فإذاً الأصناف قوالب للكائنات ويعتبر الكائن حالة instance للصنف.

كيف نستطيع توليد الصنف؟

كيف نستطيع توليد الكائن؟

## Class Syntax in C++

```
class class_name {  
    public:  
    public members declarations  
};
```

```
class MyClass {  
    public:  
    int myNum;  
    string myString; };
```

```
MyClass myObj;  
myObj.myNum = 15;
```

يبدأ تعريف الصنف بالكلمة المفتاحية class ثم اسم الصنف وهو اسم نوع معطيات جديد معرف من قبل المستخدم. والعناصر ضمن الصنف تسمى خصائص attributes.

public كلمة مفتاحية تحدد إمكانية الوصول إلى أعضاء الصنف وتسمى محدد وصول access specifier الأعضاء الموجودة في القسم العام يمكن الوصول إليها من خارج الصنف

يمكن أن تكون خصائص الصنف: توابع و متحولات

يتم توليد الكائن object من الصنف عندها يصبح له مكان في الذاكرة والوصول إلى خصائص الصنف يتم باستخدام معامل ".".

## Struct in C++: 1<sup>st</sup> example

```
#include <iostream>
using namespace std;
struct Person
{int citizenship;
int age;
};
int main() {
Person p;
p.citizenship = 1;
p.age = 27;
cout << "Person citizenship: " << p.citizenship << endl;
cout << "Person age: " << p.age << endl;
return 0;
}
```

## Struct in C++: 2<sup>nd</sup> example

```
#include <iostream>
using namespace std;
struct Length
{int meters;
float centimeters; };
int main()
{Length *ptr, l;
ptr = &l;
cout << "Enter meters: ";
cin >> (*ptr).meters;
cout << "Enter centimeters: ";
cin >> (*ptr).centimeters;
cout << "Length = " << (*ptr).meters << " meters " << (*ptr).centimeters
<< " centimeters";
return 0;}
```



## Classe : 1<sup>st</sup> example

```
#include <iostream>
using namespace std;
class MyClass {
public:
    int myNum;
    string myString;
};
int main() {
    MyClass myObj;

    myObj.myNum = 15;
    myObj.myString = "Some text";

    cout << myObj.myNum << "\n";
    cout << myObj.myString;
    return 0;
}
```

## Classe : 2<sup>nd</sup> example

```
#include <iostream>
using namespace std;
// Create a Car class with some attributes
class Car {
public:
    string brand;
    string model;
    int year; };
```

```
int main() {
// Create an object of Car
Car carObj1;
carObj1.brand = "BMW";
carObj1.model = "X5";
carObj1.year = 1999;
// Create another object of Car
Car carObj2;
carObj2.brand = "Ford";
carObj2.model = "Mustang";
carObj2.year = 1969;
// Print attribute values
cout << carObj1.brand << " " << carObj1.model << " " <<
carObj1.year << "\n";
cout << carObj2.brand << " " << carObj2.model << " " <<
carObj2.year << "\n";
return 0;
}
```

## Classe : Data and Functions: 3<sup>rd</sup> example

```
#include <iostream>
#include <iomanip>
using namespace std;
class Square {
public:
    double length;
    double width;
    double calculateArea()
    {return length * width;}
    double calculatePerimeter()
    {return 2 *(length + width);}
    void printdimensions (double, double)
    {cout << length<< setw(16)<< width;}
};
```

```
int main ()
{//Create a new Square, square1
Square square1;
//assign values to the data members
square1.length = 20.2;
square1.width = 10.5;
//Calculate the area and perimeter
cout <<"Area is: " << square1.calculateArea()<<endl;
cout <<"Perimeter is: " << square1.calculatePerimeter() <<endl;
square1.printdimensions(20.2,10.5);
return 0;
}
```

## QUIZ

عرف مصفوفة ثنائية مربعة صحيحة بعدها 3 ثم قم بتخزين عناوين عناصر قطرها الرئيسي في مصفوفة مؤشرات مناسبة ثم اطبع محتويات مصفوفة المؤشرات الناتجة

```
#include <iostream>
using namespace std;
int main()
{int myarray[3][3] = {{2,4,5},{6,8,88},{10,11,99}};
int *ptr[3]; //array of pointers
for(int i=0;i<3;i++)
    for(int j=0;j<3;j++)
        if (i==j)
            {ptr[i] = &myarray[i][j];}
for (int i = 0; i < 3; i++)
    { cout << ptr[i] << endl; }
return 0;
}
```

انتهت تمارين الأسبوع 4

جامعة  
المنارة  
HANARA UNIVERSITY