المحاضرة الثالثة

برمجة الكائنات OOP

Functions with Empty Pararmeter Lists

دوال بدون وسيطات :

في لغة ال C++ تكتب الدالة التي لا تمتلك وسيطات إما بكتابة void بين القوسين الذين يتبعان اسم الدالة أو تركهما فارغين ، فمثلاً الإعلان

```
void print ( );
```

يشير إلى أن الدالة Print لا تأخذ أي وسيطات وهي لا ترجع قيمة.

```
#include <iostream.h>

main( )
{
f1 ( );
f2 ( );
}

void f1 ( )
{
cout << "Function f1 takes no arguments" << endl;
}
void f2 ()
{
cout << "Function f2 also takes no arguments" <<
endl;
}
```

الخرج من البرنامج

```
Function f1 takes no arguments
Function f2 also takes no arguments

--------------------------------------------------------------------
```

```
main( )
{
product ( ) ;
}
void product ( )
{
int a, b, c, result;
cout << " Enter three integers: ";
cin >> a>> b >>c;
result = a*b*c;
cout << "Result is : " << result;
}
```

الفئات Classes

أساس البرامج المكتوبة باللغة C++ هو الكائنات التي يتم إنشاؤها بواسطة فئة تستعمل كقالب فعندما يكون هنالك الكثير من الكائنات المتطابقة في البرنامج لا يكون منطقياً وصف كل واحد منها على حدة ، من الأفضل تطوير مواصفات واحدة لكل من هذه الكائنات وبعد تحديد تلك المواصفات يمكن استخدامها لإنشاء قدر ما نحتاج إليه من الكائنات تسمى مواصفات إنشاء الكائنات هذه في OOP فئة **(Class)** تتميز الفئة في C++ .

/اسم الفئة والذي يعمل كنوع البيانات الذي ستمثله الفئة.

/مجموعة من الأعضاء البيانية في الفئة **(data members)** حيث يمكن أن تحتوى الفئة على صفر أو أكثر من أي نوع من أنواع البيانات في **C++**

/مجموعة من الأعضاء الدالية **(member functions)** معرفة داخل الفئة وهي تمثل مجموعة العمليات التي سيتم تنفيذها على كائنات الفئة.

/محددات وصول **(access specifiers)** وتكتب قبل الأعضاء البيانية والأعضاء الدالية لتحدد إمكانية الوصول إلى هذه الأجزاء من الأجزاء الأخرى في البرنامج.

تعريف الفئة The Class Definition

يتألف تعريف الفئة من الكلمة الأساسية **class** يليها اسم الفئة ثم جسم الفئة بين قوسين حاصرين { }ويجب أن ينهي تعريف الفئة فاصلة منقوطة أو عبارة إعلان عن كائنات تنتمي إلى الفئة فمثلاً:

```
class anyclass { /* class body*/ };
```

غالباً ما تكتب الفئة في C++ على النحو التالي في البرنامج .

```
class class_name{
private:
data members
public:
member functions
};
```

المثال التالي يوضح كيفية تعريف فئة

```
// This creates the class stack >
class stack {
private:
int stck[SIZE];
int tos;
public:
void init ( );
void push(int i);
int pop ( );
};
```