

lesson 7 CPP Set

في الدروس السابقة تعلمنا كيف نقوم بترتيب عناصر **Array** و **Vector** باستخدام **sort** لكن في هذا الدرس سنتعرف على **set** تعمل **set** نفس عمل **vector** من نفس المميزات تكبير حجمها أو تصغيرها لكن الميزة الموجودة بـ **set** هو ترتيب العناصر تلقائياً ولكن هناك عيب بها وهو أنها تقوم بحذف العناصر المكررة و نقوم بتعريفها عن طريق تضمين مكتبة تسمة **set** ونقوم بتعريفها مثلها مثل **vector** وتكون كالتالي :

```
set <data type> name_of_set;
```

عندما نريد إدخال عنصر نقوم بكتابة **insert()** كالتالي :

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

```
int main() {
    set <int> x;
    x.insert(9);
    x.insert(1);
    x.insert(4);
    x.insert(3);
    x.insert(9);
}
```

هنا قمنا بإدخال قيم العناصر بترتيب مختلف وعناصر مكررة

لطباعة عناصر **set** ، هنا يكون الأمر مختلف عن طباعة عناصر **vector**
نقوم بكتابة **set** ثم يليها **< >** وبداخلها نوع **data type** ثم يليها **::** ثم **iterator** وهنا
هو مؤشر يشير إلى مكان **set**
ثم نقوم بتسميته لل **iterator**
ثم نقوم بعمل **for** لنحدد بداية **set** ونهايته وهنا المؤشر يشير الى كل عنصر في **set**
وقمنا هنا بتسميته **it** لكن يمكنك تسميته اى اسم تريده
والطباعة تكون بهذه الطريقة

```
set <int> ::iterator it;  
for (it = x.begin(); it != x.end(); it++)  
    cout << (*it) << endl;
```

مثال :

```
#include <iostream>  
#include <set>  
using namespace std;  
int main() {  
    set <int> x;  
    x.insert(9);  
    x.insert(1);  
    x.insert(4);  
    x.insert(2);  
    x.insert(3);  
    x.insert(9);  
    x.insert(4);  
    set <int> ::iterator it;  
    for (it = x.begin(); it != x.end(); it++)  
        cout << (*it) << " ";  
}
```

output:

1 2 3 4 9

قامت بطباعتها بالترتيب وحذف العناصر المكررة