

جامعة المنارة

مقرر برمجة 2

د. هلا نصار

م. أوشين داود

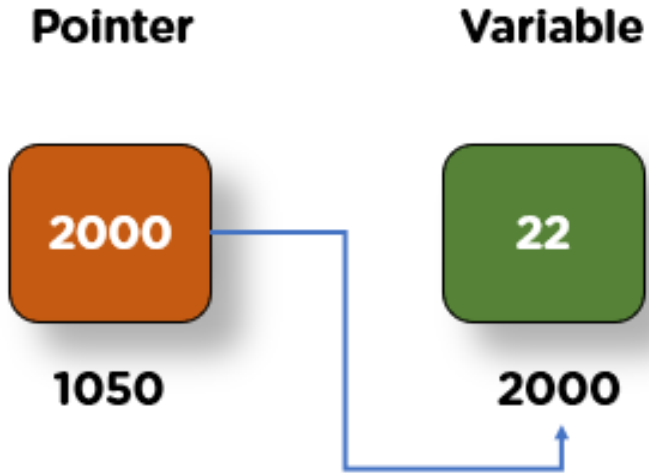
جامعة
المنارة

HANARA UNIVERSITY

محاضرات الأسبوع 1

الفصل الأول 2021-2022

المؤشرات Pointers



لكي نفهم المؤشرات لا بد من فهم كيفية تخزين المعطيات في الحاسب.
عند التصريح عن متحول variable من نوع ما:

يتم حجز مكان له في الذاكرة حجمه يتناسب مع نوع المتحول، هذا المكان له عنوان، ويتم تخزين قيمة المتحول في هذه المكان المحجوز. للوصول إلى عنوان هذا المكان يستخدم المعامل & قبل اسم المتحول ويسمى معامل المرجع (reference).

وبالمقابل عند التصريح عن مؤشر من نوع ما:

يتم حجز مكان في الذاكرة يتسع لعنوان، هذا العنوان يجب أن يكون لمكان في الذاكرة حجمه يتناسب مع نوع المؤشر.

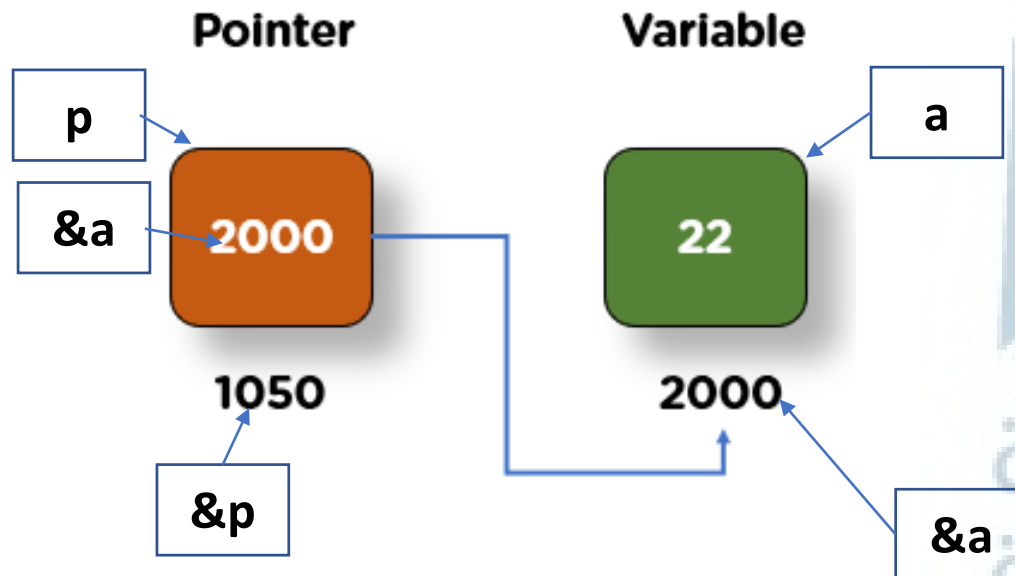
مثلاً إذا كان لدينا متحول من النوع int سيتم حجز مكان في الذاكرة 4 byte أما المؤشر من النوع int سيتم حجز مكان يتسع لعنوان ذاكرة حجمها 4 byte. فإذا أعطي عنوان المتحول أصبح مؤشراً pointer له.
إذاً المؤشر هو متحول يحتوي عنواناً لمتحول آخر فيشير إليه.

التصريح عن مؤشر Pointer Declaration Syntax

```
datatype *pointer_name;
```

يسمى المعامل *: Deference operator

```
int *x; // a pointer to integer  
double *x; // a pointer to double  
float *x; // a pointer to float  
char *ch; // a pointer to a character
```



تهيئة مؤشر Pointer initialization

```
int a = 22 ;  
int *p = &a;
```

Reference operator (&) and Deference operator (*)

```
#include <iostream>
using namespace std;
int main()
{
int x = 27;
int *p; // declaring a pointer
p = &x; // assign the address of x to p
cout << x << endl; // Print the value of variable x
cout<<&x<<endl; //Print the address of variable x
cout << "Value of p is : "; cout << p<< endl;
cout << "Value of *p is : "; cout << *p << endl;
cout << "Value of &p is : ";
cout << &p<<endl; // Print the address of variable p
return 0;
}
```

Reference operator (&) and Deference operator (*)

```
#include <iostream>
using namespace std;
int main()
{
int x = 27;
int *p; // declaring a pointer
p = &x; // assign the address of x to p
int &z = x;
cout << x << endl; // Print the value of variable x
cout << &x << endl; // Print the address of variable x
cout << z << endl; // Print the value of variable z
cout << &z << endl; // Print the address of variable z
cout << "Value of p is : "; cout << p << endl;
cout << "Value of *p is : "; cout << *p << endl;
cout << "Value of &p is : ";
cout << &p << endl; // Print the address of variable p
return 0;
}
```

Pointer Arithmetic العمليات الحسابية على المؤشرات

بما أن المؤشرات متحويلات قيمها عناوين، أي قيم عددية، فيمكن إجراء بعض العمليات الحسابية على المؤشر لتغيير قيمته وفيما يلي العمليات الأربع الأساسية المطبقة على المؤشرات:

- معامل الزيادة ++ increment operator
- معامل الإنقاص – decrement operator
- الجمع + addition
- الطرح – subtraction

لكن لا بد من التمييز بين تطبيق العمليات على المؤشر وتطبيقها على قيم ما يشير إليه.

-1 increment operator :

عند استخدام المعامل مع مؤشر فإنه يقوم بزيادة قيمة المؤشر بمقدار حجم نوع المعطيات الذي يشير اليه المؤشر. إن المؤشر p يشير الى متحول من النوع int والذي حجمه في الذاكرة يساوي 4 byte عند تطبيق المعامل تزداد قيمة المؤشر وهي عنوان بمقدار 4 byte أي يصبح مؤشر لمكان في الذاكرة يلي الموقع السابق.

```
#include <iostream>
using namespace std;
int main()
{
int a =30;
int *p = &a;
cout<<*p<<endl<<a<<endl;
(*p)++;
cout<<*p<<endl<<a<<endl;
cout << p<<endl ;
p++;
cout <<p<<endl;
return 0;
}
```

ملاحظة

ان وجود الاقواس في تعليمة `(*p)++` يعطي الأولوية لمعامل * على معامل الزيادة ++ من اجل فهم ذلك: استبدل السطر المشار اليه في البرنامج بالحالات التالية ولاحظ الخرج:

1. `*p++`
2. `*++p`
3. `++*p`

decrement operator -2

عند استخدام المعامل مع مؤشر فإنه يقوم بإنقاص قيمة المؤشر بمقدار حجم نوع المعطيات الذي يشير اليه المؤشر. إن المؤشر p يشير الى متحول من النوع int والذي حجمه في الذاكرة يساوي 4 byte عند تطبيق المعامل تنقص قيمة المؤشر بمقدار 4 byte أي يصبح مؤشر لمكان في الذاكرة يسبق الموقع الحالي.

```
#include <iostream>
using namespace std;
int main()
{
int a =30;
int *p = &a;
cout<<*p<<endl<<a<<endl;
(*p)--;
cout<<*p<<endl<<a<<endl;
cout << p<<endl ;
p--;
cout <<p<<endl;
return 0;
}
```


3- معامل الجمع +n

عند استخدامه مع مؤشر فإنه يقوم بزيادة قيمة المؤشر بمقدار عدد من المرات n * حجم نوع المعطيات الذي يشير اليه المؤشر

إن المؤشر p يشير الى متحول من النوع `int` والذي حجمه في الذاكرة مساو الى `4 byte` عندما نكتب `p+3` يتم الانتقال بالذاكرة بمقدار 12 بايت الى موقع يلي الموقع الحالي

```
#include <iostream>
using namespace std;
int main()
{
int a =30;
int *p = &a;
cout << p<<endl ;
cout << p+3<<endl;
cout<< *p<< endl;
*p +=3;
cout<< *p<< endl;
return 0;
}
```

4- معامل الطرح n:-

عند استخدام المعامل مع مؤشر فإنه يقوم بإنقاص قيمة المؤشر بمقدار عدد من المرات n * حجم نوع المعطيات الذي يشير اليه المؤشر.
إن المؤشر p يشير الى متحول من النوع int والذي حجمه في الذاكرة مساو الى 4 byte عندما نكتب $p-3$ يتم الانتقال بالذاكرة بمقدار 12 بايت الى موقع سابق.

```
#include <iostream>
using namespace std;
int main()
{
int a =30;
int *p = &a;
cout << p<<endl ;
cout << p-3<<endl;
cout<< *p<< endl;
*p -=3;
cout<< *p<< endl;
return 0;
}
```

```
#include <iostream>
using namespace std;
int main()
{
int a =30;
int *p = &a;
cout << p<<endl ;
p=p-3;
cout<< *p<< endl;
*p -=3;
cout<< *p<< endl;
return 0;
}
```

ملاحظة :
انتبه الى الخرج في
البرنامج التالي

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

int main() {

    int num1, num2, *ptr1, *ptr2, sum=0;
        cout<<"Enter the two number :";
        cin>>num1>>num2;
        ptr1 = &num1;
        ptr2 = &num2;
        sum = *ptr1 + *ptr2;
        cout<<"Sum of the two number is "<<sum;

    return 0;

}
```

تذكرة : النظام العددي الست عشري Hexadecimal numbering system

0 , 1, 2, 3, 4, 5, 6,7,8,9, A,B,C,D,E,F :

A = 10, B = 11, C = 12, D = 13, E = 14, F = 15.



QUIZ1

اكتب برنامج لطباعة الأكبر بين عددين صحيحين باستخدام المؤشرات

```
#include <iostream>
using namespace std;
int main()
{ int num1, num2, *ptr1, *ptr2;
  cout<<"Enter the two number :";
  cin>>num1>>num2;
  ptr1 = &num1;
  ptr2 = &num2;
  if (*ptr1 > *ptr2)
    { cout << *ptr1; }
  else
    { cout << *ptr2; }
  return 0;
}
```

QUIZ2

اكتب برنامج لاختبار عدد مدخل صحيح هل هو زوجي أم فردي باستخدام المؤشرات

```
#include <iostream>
using namespace std;
int main()
{int * ptr1; int num1;
cout<<"Enter the number :";
cin>>num1;
ptr1 = &num1;
if ((*ptr1)%2==0)
    cout << "it is even";
else
    cout << "it is odd";
return 0;}
```

انتهت تمارين الأسبوع الأول



جامعة
المنارة

HAMARA UNIVERSITY