

دكتور الملاءة: سمير جعفر

المحاضرة: الخامسة

نظري

المستوى العلمي: أهلاً بكم أصدقائي في المحاضرة الخامسة من مقرر الخوارزميات والبرمجة 2، في هذه المحاضرة سوف نكمل مع تمرين في المؤشرات، إضافة لتعليمتي (new, delete).

تمرين: اكتب برنامج بلغة C++ يقوم بإدخال عددين صحيحين ومن ثم تبديل محتويات المتحولين وطباعتهما وذلك باستخدام المؤشرات (الحل مرة باستخدام متحول ثابت ومرة دون متحول ثابت).

الحل:

```
#include < iostream.h >
```

عرفنا ثلاث متغيرات من النوع الصحيح المتغير الثالث ليساعدنا في عملية التبديل

```
int main( ){ int x, y, z;
```

التصريح عن ثلاث مؤشرات

```
int * a,* b,* d;
```

إدخال العددين الصحيحين

```
cin >> x >> y;
```

المؤشر a يحمل عنوان المتحول x

```
a = &x;
```

المؤشر b يحمل عنوان المتحول y

```
b = &y;
```

المؤشر d يحمل عنوان المتحول z

```
d = &z;
```

```
* d =* a;
```

```
* a =* b;
```

```
* b =* d;
```

اسندنا محتوى الخانة التي يؤشر عليها d الى محتوى الخانة التي يؤشر عليها d اسندنا قيمة x ل z أي ان z لها قيمة x

اسندنا محتوى الخانة التي يؤشر عليها b الى محتوى الخانة التي يؤشر عليها a (اسندنا قيمة x ل y أي ان x لها قيمة y)

اسندنا محتوى الخانة التي يؤشر عليها d الى محتوى الخانة التي يؤشر عليها لا (اسندنا قيمة y ل z أي ان y لها قيمة z)

```
cout << x << y;
```

```
return 0; }
```

الحل دون استخدام متحول ثابت : (دون استخدام متحول ثالث z كما في الطريقة الأولى):

```
#include < iostream.h >
```

```
int main (){
```

```
int x , y ;
```

```
cin >> x >> y;
```

```
int * b,* d;
```

```
* b = &x;
```

```
* d = &y;
```

```
* b =* b +* d;
```

```
* d =* b -* d;
```

```
* b =* b -* d;
```

```
cout << x << y;
```

```
return 0; }
```

اسندنا قيمة مجموع العددين الى محتوى الخانة المؤشر عليها ب b (قيمة x الان هي مجموع العددين)

اسندنا قيمة جمع محتوى الخانتين المؤشر عليهم ب b و d الى محتوى الخانة المؤشر عليها ب d (قيمة y الان هي قيمة طرح y من قيمة x الجديدة (مجموع العددين)) أي ان قيمة y الان هي قيمة x

اسندنا قيمة قيمة جمع محتوى الخانتين المؤشر عليهم ب b و d الى محتوى الخانة المؤشر عليها ب b (قيمة x الان هي قيمة طرح x الجديدة (مجموع العددين) من قيمة y الجديدة (طرح العددين)) أي ان قيمة x الان هي قيمة y

تعلمنا سابقاً أصدقائي :

- التصريح عن متحول من نوع معين :
- إعطاء قيمة للمتحول :
- التصريح عن متحول من نوع مؤشر :
- الحصول على عنوان الخانة في الذاكرة التي خصصت للمتحول x باستخدام معامل العنوان (&).
- الحصول على محتوى الخانة المؤشر عليها باستخدام معامل المحتوى (*).

```
int x;
```

عندما كتبنا :

صرحنا عن متحول من نوع صحيح وبنفس الوقت حجزنا صندوق(خانة) في الذاكرة باسم المتحول x

➤ **لكن** : ماذا إذا أردنا أن نحجز مكان في الذاكرة لمتحول دون التصريح عنه أو إعطاؤه اسم !!!!!!!

نستطيع ذلك باستخدام التعليمات *new* التي تعني حجز مكان جديد في الذاكرة .

➤ لنستخدم التعليمة *new* علينا أن نحدد السعة (طول الخانة) التي نريد حجزها (تتسع لقيم صحيحة أو حقيقية أو محارف.....).

➤ لذلك علينا تحديد النوع الذي نريد حجز مكان له في الذاكرة : النوع ; *new*

➤ مثلاً: حجزنا مكان في الذاكرة يتسع لمتحول صحيح : *new int;*

كيف نستطيع استخدام هذا الصندوق الذي حجزناه دون اسم؟؟

إن تعليمة *new* ستقوم بإرجاع عنوان الصندوق (الخانة) الذي حجزناه ولأنها سترجع عنوان لذلك يجب أن يكون المتحول الذي سيحمل القيمة المرجعة (العنوان) من نوع مؤشر .

لذلك نكتب : $int * p = new int;$

أي حجز مكان في الذاكرة ليس له اسم ولم نصرح عنه مسبقاً بل له فقط مؤشر يؤشر عليه لنستطيع استخدامه.

$int * p = new int ;$ ➤

لو أردنا إعطاء قيمة 5 للصندوق المحجوز: نستخدم معامل المحتوى (*) ونكتب : $* p = 5 ;$

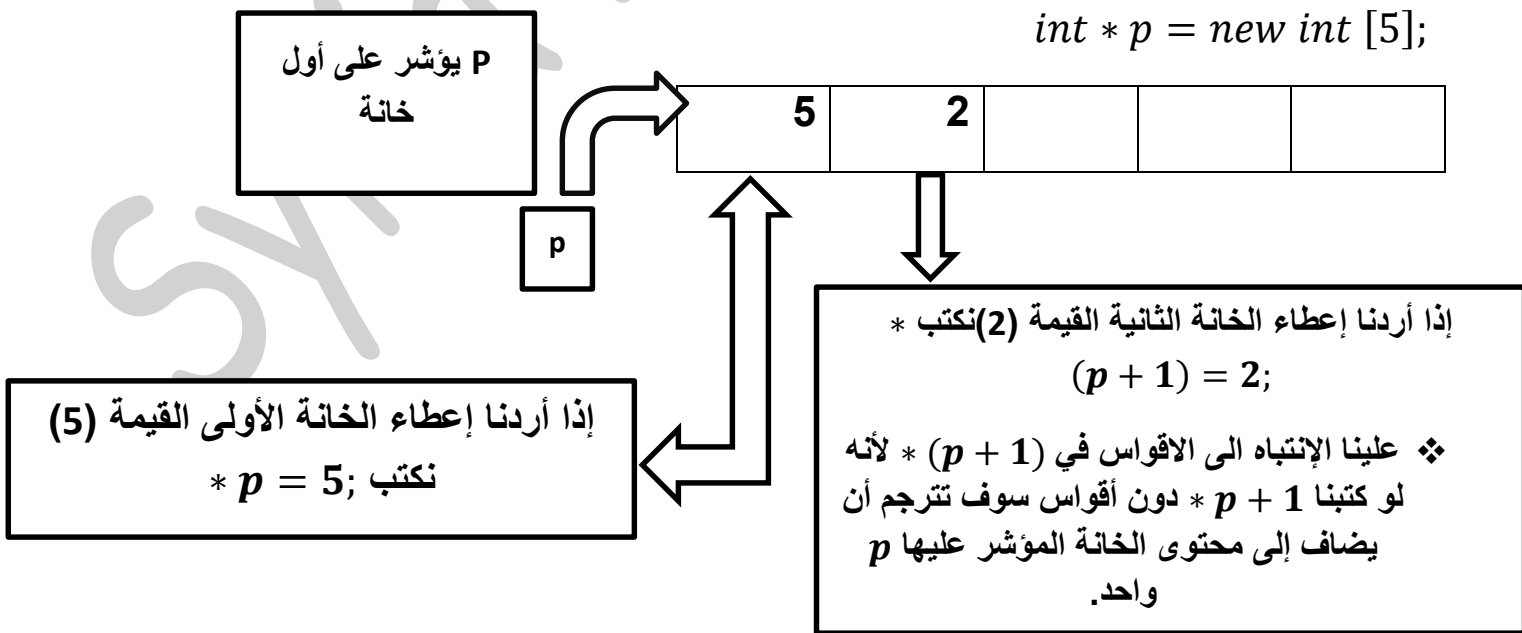
ولو أردنا زيادة واحد للقيمة المسبقة (5) في الصندوق نكتب : $* p ++ ;$

والآن نريد حجز خمس أماكن متجاورة في الذاكرة تتسع لنوع صحيح !!!?

$new int [5]$

نكتب

$int * p = new int [5];$



تعريف متجهة أعداد صحيحة بعدها (5):

`int A[5];`

- هنا صرحنا عن المتجهة وحجزنا في الذاكرة خمس أماكن متجاورة .
- إن اسم المتجهة A حقيقة هو مؤشر على متحول من النوع الصحيح وهو يؤشر على أول خانة من المتجهة .

$$* A = A[0] \rightarrow A = \&A[0]$$



- ❖ **وكما لدينا** حجز للذاكرة باستخدام `new` يمكننا حذف الحجز باستخدام التعليمة `delete` (لانقصد بالحذف حذف صندوق من الذاكرة وإنما تحرير مكان في الذاكرة (حذف المحتوى)).
- ❖ نستخدم تعليمة `delete` كالاتي: `delete p;`
- ❖ الكلمتان `new`, `delete` كلمتان محجوزة في لغة ++ c لذلك تكتب باحرف صغيرة .

تمرين:

باستخدام المؤشرات اكتب برنامج يقوم بإدخال متجهة أعداد حقيقية من البعد (20) ثم طباعتها بشكل عكسي .

الحل:

```
#include <iostream.h >
```

```
int main (){
```

```
new double[20];
```

حجز (20) صندوق متجاورين كل صندوق يتسع لعدد حقيقي

```
double * d = new double [20];
```

عرفنا مؤشر d ليحمل القيمة المرجعة (عنوان أول صندوق محجوز)

```
for (int i = 0; i < 20; i ++)
```

إدخال عناصر المتجهة ($i=0$ لأن دليل المتجهة يبدأ من الصفر)

```
cin >>* (d + i);
```

```
for(int j = 19; j ≥ 0; j --)
```

الطباعة بشكل معكوس بدأنا من القيمة 19 لننتهي بالقيمة صفر

```
cout <<* (d + j) << endl;
```

endl لطباعة كل عنصر
على سطر

```
return 0; }
```

التمارين المباشرة

إعداد: عيس خزنته كاتبي - سندس درويش - وفاء شيخ سالم