

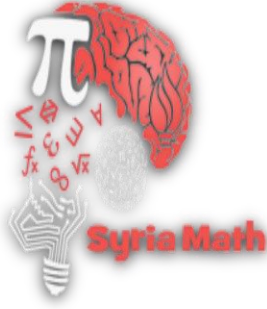
23-10-2017

نظري

دكتور الملاءة: سمير جعش

عنوان المحاضرة: المؤشرات

المحاضرة: الثالثة والرابعة



المستوى العلمي: أهلاً بكم أصدقائي في المحاضرتين الثالثة والرابعة من مقرنا الخوارزميات والبرمجة (2)، تتضمن المحاضرة الثالثة حل تمرين عن السجلات، أما المحاضرة الرابعة سنبدأ بمفهوم جديد وهو المؤشرات. لنبدأ معاً:

حل تمرين المحاضرة السابقة: اكتب برنامج بلغة C++ يقوم بتعريف سجل عامل يحتوي على البيانات التالية: اسم العامل، عمره، راتبه، جنسه والمطلوب:

1- إدخال سجلات n عامل حيث n عدد طبيعي مدخل.

2- طباعة عدد العمال الذكور.

3- طباعة سجل العامل الذي يتقاضى أدنى راتب.

الحل: سنعرف سجل يحوي على حقول مناسبة للمعلومات المذكورة في نص السؤال ثم إنشاء دالة لإدخال سجل عامل ودالة لطباعة سجل عامل ثم استخدام التعليمات والشروط المناسبة للحل.

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

```
struct factory {
```

```
string name ;
```

```
int age ;
```

```
int many ;
```

```
bool kind ;};
```

```
factory inbut(){
```

```
factory f ;
```

```
cout << "name:"; cin >> f.name;
```

```
cout << "age:"; cin >> f.age;
```

التصريح عن سجل عامل بالحقول المطلوبة

التصريح عن دالة الادخال ونوع ارجاعها
من نوع السجل

```
cout << "many: "; cin >> f.many;
```

```
cout << "kind";
```

```
char c; cin >> c;
```

```
if (c == 'm') f.kind = true; else f.kind = false;
```

```
return f; }
```

ارجاع قيمة (معلومات) العامل لدالة الإدخال

```
void print (factory f){
```

```
cout << f.name << f.age << f.many;
```

```
if(f.kind == true)cout << "male"; else cout<<"female"; }
```

التصريح عن دالة
الطباعة

```
int main (){
```

الدالة الرئيسية

```
factory A[100];
```

متجهة العمال اخترنا 100 كحد أقصى

```
int n;
```

```
do{cin >> n;}while(n > 100||n ≤ 0);
```

لضمان إدخال عدد موجب أصغر من 100

```
for(int i = 0, i < n, i ++)
```

```
A[i] = input();
```

إدخال عناصر المتجهة عن طريق
استدعاء دالة الإدخال

```
int s = 0; for(int i = 0, i < n, i ++)
```

```
if (A[i].kind == true)s ++;
```

```
cout << s;
```

```
factory min;
```

```
min = A[0]
```

```
for(int i = 1; i < n; i ++)
```

```
if (A[i].many < min.many)min = A[i];
```

عرفنا متغير من نوع السجل ليأخذ بيانات سجل العامل الأول وحلقة *for* لمقارنة رواتب باقي العمال مع راتب العامل الأول (ابتدأنا عدد الحلقة من القيمة 1 كي لا يقارن راتب العامل الأول مع نفسه) ثم تعليمة الاختبار *if* أي كلما وجدت راتب عامل أصغر من راتب العامل الأول المخزن في المتغير *min* أعطت قيمة الراتب الأصغر الى المتغير *min* وخزنت دليله في المتغير *i*

```
print (min);
```

طباعة سجل العامل الذي يتقاضى ادنى راتب

```
return 0; }
```

المحاضرة الرابعة:

مقدمة: تكمن أهمية المؤشرات في لغة ++C بإعطائها اللغة فاعلية وسرعة أكبر ولكن المؤشرات موضوع صعب يتطلب الحذر والدقة أثناء استخدامها في البرنامج لتفادي الأخطاء لذلك أغلبية اللغات لا تستخدمها .

❖ إن الهدف من المؤشرات التعامل مع الذاكرة بشكل مباشر.

¶ عند التصريح عن متحول من النوع الصحيح نكتب:

حجز خانة في الذاكرة سميت باسم المتحول x :
 $int x$;
 أعطيناها القيمة (3)
 $x = 3$;

¶ ذاكرة الحاسب: عبارة عن خانات متجاورة حجم كل منها واحد بايت كل خانة من خانات الذاكرة مرقمة برقم من النظام الست العشري ويسمى عنوان هذه الخانة .

¶ المؤشر: متحول (متغير) يحمل عنوان متحول آخر (وهي قيمة عددية مكتوبة بالنظام الست عشري).

¶ التصريح عن متحول من نوع مؤشر:

; اسم المؤشر * نوع المتحول المؤشر عليه

مثال: لنصرح عن متحول يؤشر على متحول من نوع صحيح:

$int * p$;

(p) متحول يؤشر على متحول من نوع صحيح (مؤشر على نوع صحيح) وقيم المتحول (p) عبارة عن عناوين خانات حجزت في الذاكرة تتسع لمتحول من نوع صحيح (المتحول من النوع الصحيح يحجز 2byte).

إن نوع المتحول يتحكم بطول الخانة (حجمها) (2byte, 4byte,).

مثال: للتصريح عن مؤشر يؤشر على متحول من النوع المحرفي: $char * q$;

$int x = 3$;

¶ لدينا:

$int * p$;

السؤال: كيف نستطيع الحصول على عنوان خانة في الذاكرة المخصصة للمتحول x ؟؟
 باستخدام معامل العنوان (&).

مثال:

```
int x = 3;
```

```
int * p;
```

```
p = &x;
```

(p) يُوْشِر على متحول من نوع صحيح , $p = \&x$ قيمة p هي عنوان الذاكرة للخانة x وهي رقم ست عشري .

❖ للوصول الى العنوان إما تكتب (p) أو ($\&x$) الاثنان يدلان على نفس القيمة .

توضيح: ماخرج تنفيذ المقطع البرمجي التالي :

```
int x = 3;
```

```
int * p;
```

```
p = &x;
```

```
cout << p << &x;
```

سيطبعان نفس
القيمة

33

```
int x = 3;
```

```
int * p;
```

```
p = &x;
```

```
int f;
```

```
char * c;
```

```
c = &f;
```

❖ بعد تعرفنا على معامل العنوان نعود لفكرة بسيطة :

المؤشر p من نوع صحيح ويُوْشِر على نوع صحيح .

➤ هنا علينا الانتباه دائما لأن نوع المؤشر يجب أن يكون

نفس نوع المتحول الذي يُوْشِر عليه .

➤ **مثلاً:**

هذا خاطئ لأنه لايمكن استخدام مؤشر من نوع محرفي ليُوْشِر

على متحول من نوع صحيح .

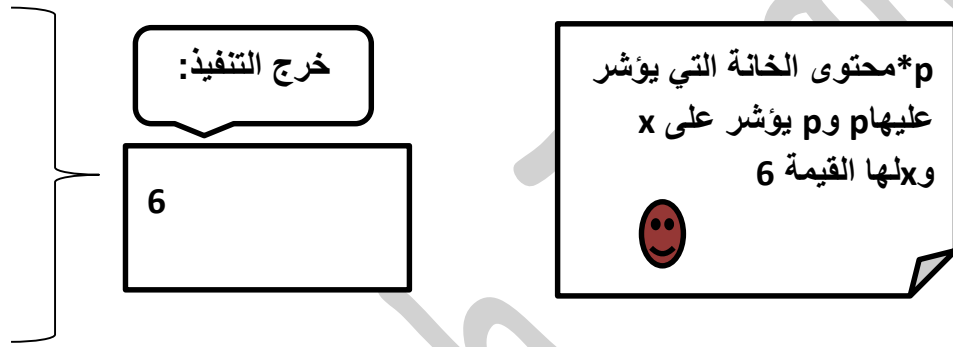
سؤال: كيف نستطيع الوصول الى محتوى (قيمة) الخانة المؤشر عليها ؟؟؟؟

نستطيع الوصول الى محتوى أي خانة مؤشر عليها باستخدام معامل المحتوى (*).

- عندما نستخدم معامل المحتوى مع مؤشر هذا يعني اننا نحصل على محتويات الخانة المؤشر عليها بهذا المؤشر .

مثال: ماخرج تنفيذ المقطع البرمجي التالي :

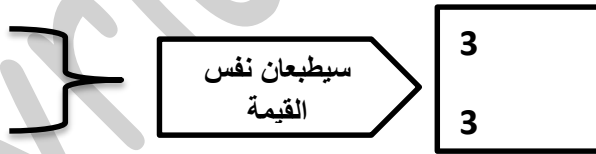
```
int * p ;
int x ;
p = &x;
* p = 6;
cout << x;
```



❖ للوصول إلى محتويات الخانة نكتب (*p) أو (x) لهما نفس القيمة .

توضيح: ماخرج تنفيذ المقطع البرمجي التالي :

```
int x = 3;
int * q;
q = &x;
cout << x;
cout << * q;
```



❖ **ملاحظة:** بالنسبة ل (*) لها عدة استخدامات:

1- (*) من العمليات الحسابية نستخدمها لعملية الضرب .

2--معامل محتوى .

3-التصريح عن مؤشر. للتمييز بين معامل المحتوى والتصريح عن مؤشر: إذا وجد قبل ال(*) نوع لمتحول (int,char....) فهو تصريح لمؤشر وإذا لم يوجد نوع متحول قبل '*' فهو عبارة عن معامل محتوى.

مثال: ماخرج تنفيذ البرنامج التالي :

```
#include < iostream.h >

int main (){

double x = 3,y = 5 ,int z = 2;

double * p,* q;

int * w;

p = &x,w = &z,q = &y ;

cout <<* q;
cout << q;
cout <<* y;
cout << y;
cout << &y;
return 0 ; }
```

خرج
التنفيذ

Cout <<*q; أي محتوى الخانة المؤشر عليها أي
محتوى y فخرج التنفيذ 5

Cout <<q; أي عنوان خانة الذاكرة المخصصة لـ y
فخرج التنفيذ 0XABFFE40

Cout <<y; فخرج التنفيذ 5

Cout <<*y; خطأ عند التنفيذ لأن y متحول صحيح
وليس مؤشر

Cout <<&y; أيضا سيكون خرج التنفيذ عنوان y في
الذاكرة 0XABFFE40

انتهت المحاضرة.....

إعداد : عبير خزنة كاتبي & سندس درويش & وفاء شيخ سالم