

دكتور الملاءة: سمير جعفر

المحاضرة: الأولى

عنوان المحاضرة: الدوال والسجلات

عملي

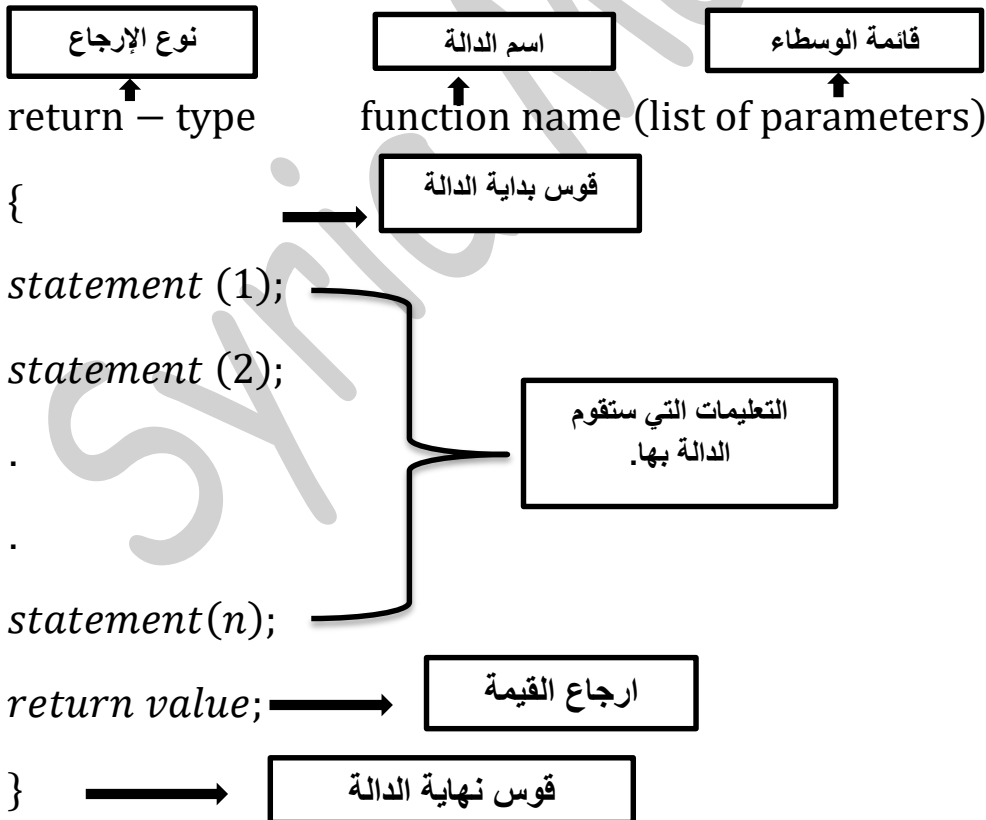
المستوى العلمي: أهلاً بكم أصدقائي في المحاضرة الأولى من مقرر العملي للخوارزميات والبرمجة (2)، تتضمن هذه المحاضرة: 1-مراجعة لمفهوم الدوال -2-تطبيق برنامجين باستخدام الدوال. 3-السجلات .

أصدقائي لقد تم التعديل على جلسات العملي وأصبحت بالشكل التالي : كل إربعاء سيتم وضع تمارين للجلسة القادمة في الطابق الرابع في لوحة الاعلان ليسجلها الطالب ويقوم بتطبيقها مباشرة في الجلسة.

البنية الأساسية للدالة:

تعريف الدالة: برنامج جزئي يقوم بعمل محدد ، والبرنامج بلغة ++C يتألف

من عدة دوال.



شروط تسمية الدوال والمتغيرات:

- لا يبدأ برقم.
- لا يحوي فراغات أو رموز.
- ليس من الكلمات المحجوزة .

يمكن كتابة الدوال داخل البرنامج بطريقتين:

- التصريح والتعريف عنها قبل الدالة الرئيسة (main).
- التصريح عنها قبل الدالة الرئيسة (باستخدام ترويسة الدالة) ومن ثم تعريفها بعد الدالة الرئيسة .

التصريح عن دالة :

بذكر نوع ارجاعها واسمها مع قائمة الوطاء .

مثلاً $int f (int x);$ وهذا ماندعوه ترويسة الدالة.

استدعاء الدالة في البرنامج :

- يمكن استدعاء الدالة التي لها نوع إرجاع غير void من خلال الطباعة مباشرة (باستخدام تعليمة cout).
- إسناد الدالة إلى متحول من نفس نوع إرجاع الدالة (بهذه الطريقة نكون قد حفظنا ماترجعه الدالة في المتحول).
- الدوال التي لاترجع أي قيمة تستدعى بذكر اسمها فقط (دالة الطباعة).

ملاحظات:

- كل ما يتم تعريفه داخل الدالة ينتهي عمله (استخدامه) مجرد الانتهاء من الداله فلا يمكن استخدامه في دالة ثانية(فلو أردنا إعادة استخدامه في دالة أخرى نعرفه قبل الدوال وليس ضمنها).
- الدوال التي نوع إرجاعها void(مثل دالة الطباعة) ليست بحاجة إلى التعليمة return أو نكتب أنها لاترجع شيء بالشكل التالي (return ;).

البرنامج الأول :

اكتب برنامج يستخدم الدوال لطباعة العبارة *****n***** حيث n عدد صحيح موجب تماما يحدد قيمته المستخدم عند التنفيذ باستخدام دالة الطباعة.

الحل: يمكن كتابة اسم المكتبة بالشكل `#include < iostream.h >` لكن دون كتابة السطر `Using name space std;`

```
#include < iostream >
Using name space std;
void print (int n)
{cout << "***" << n << " ***";}
int main( ) {
int n ;
do{cout << "n="; cin >> n; }while(n <= 0);
print(n);
return 0;}
```

استخدمنا الحلقة `do - while` للتأكد من أن الرقم المدخل هو رقم موجب تماما .

استدعينا الدالة بذكر اسمها فقط لأنها لن ترجع قيمة بل تقوم بطباعة الشكل.

```
n = 6
*** 6 ***
```

نتيجة التنفيذ تكون بالشكل :

نتيجة التنفيذ

على فرض $n = 6$

البرنامج الثاني :

اكتب دالة لحساب كل مما يلي (دالة لكل عملية):

1- حساب مضروب العدد n (العالمي $n!$).

2- حساب قواسم العدد $n >$

3- اختبار فيما إذا كان العدد أولي أم لا .

ثم اكتب برنامج يقوم بإدخال عدد وليكن n من النوع الصحيح ويقوم بتنفيذ مايلي:

1- طباعة مضروب العدد n (عالمي $n!$).

2- طباعة قواسم العدد في حال لم يكن أوليا .

وذلك باستخدام الدوال السابقة .

: الحل

```
#include < iostream >
```

```
Using name space std;
```

```
int fact(int n ){
```

```
int z = 1;
```

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

```
z = z * i;
```

```
return z ;
```

```
}
```

```
void div (int n){
```

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

```
if (n%i == 0)
```

```
cout << i << " " ;
```

```
}
```

```
int prim (int n ){
```

```
int t = 1
```

```
for (int i = 2; i < n; i ++)
```

```
if (n%i == 0)
```

```
{
```

```
t = 0
```

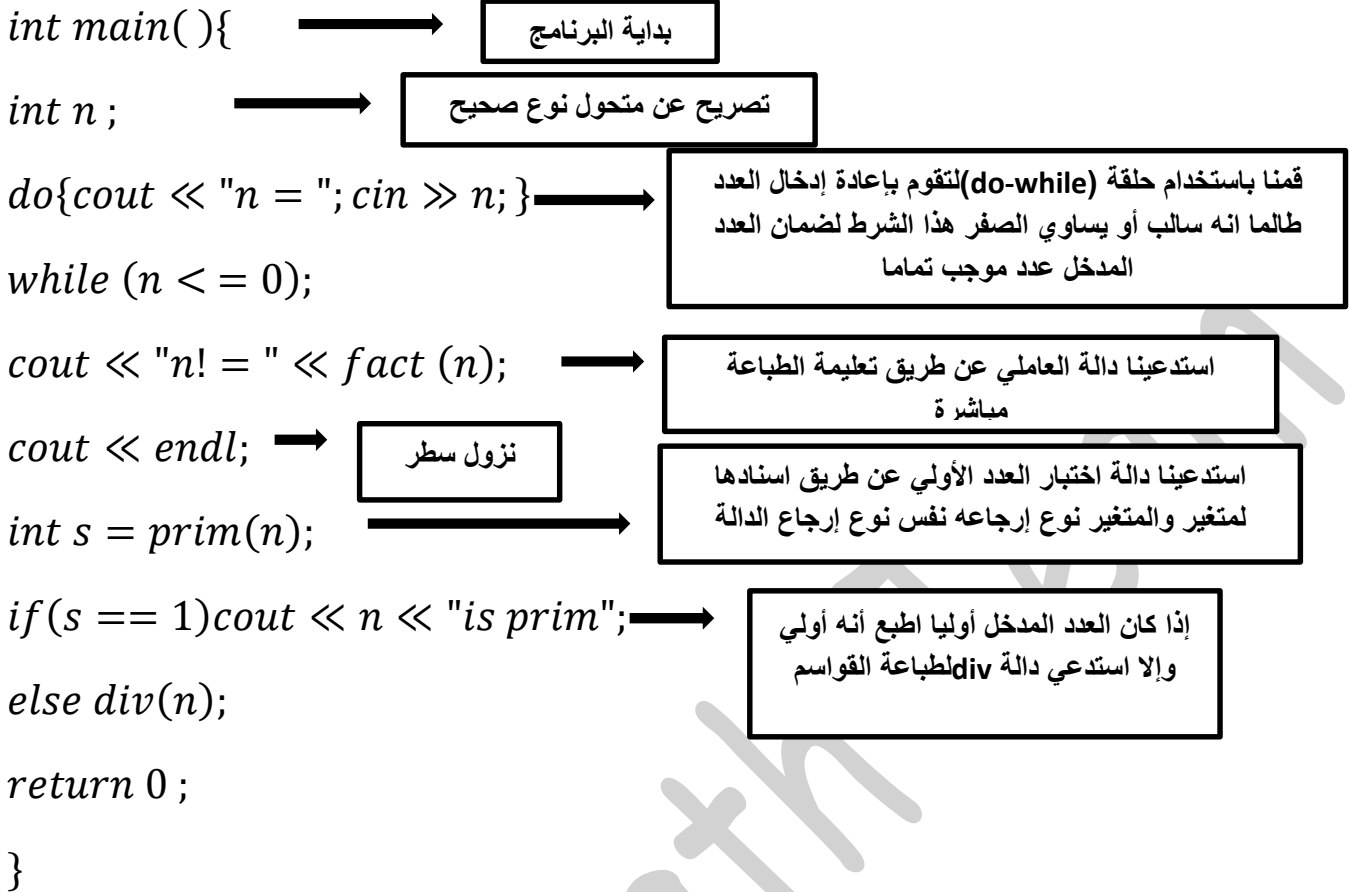
```
break; }
```

```
return t; }
```

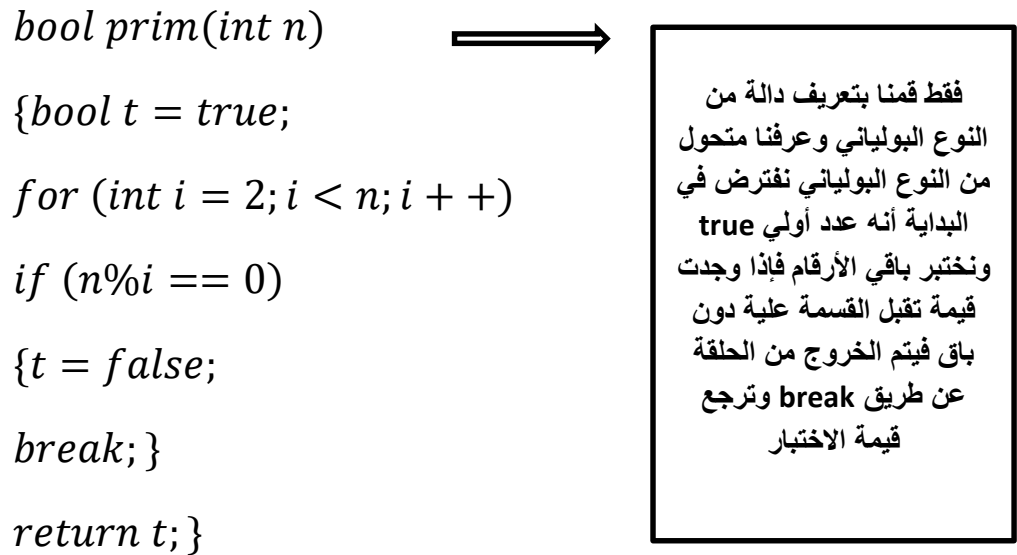
قمنا بتعريف دالة العامل نوع إرجاعها النوع الصحيح لأن العاملى دائما عدد صحيح أسمينا الدالة fact والدالة تحوي وسيط واحد لتحسب العاملى له وترجع قيمته للدالة

قمنا بتعريف دالة من النوع void لاترجع قيمة هي فقط تطبع قواسم العدد

قمنا بتعريف دالة لاختبار العدد الاولي هذه الدالة نوع إرجاعها النوع الصحيح اسميناها prim هذه الدالة تحوي وسيط واحد نفترض في البداية انه عدد أولي (t=1) ثم نقوم باختبار باقى الاعداد فمجرد أن وجد عدد يقبل القسمة عليه دون باقى (n%i==0) تخرج من الحلقة عن طريق break وتعيد قيمة الاختبار.



• طريقة ثانية لاختبار فيما إذا العدد أولي ام لا:



$n = 6$

$n! = 720$

1 2 3

نتيجة التنفيذ بالشكل :

على فرض $n = 6$

السجلات:

التعريف: هو نوع بيانات مركب يحوي على حقول غير متجانسة بالضرورة .

الشكل العام للسجلات:

struct اسم النوع {

اسم الحقل الأول نوع الحقل الأول ;

اسم الحقل الثاني نوع الحقل الثاني ;

; اسم الحقل الأخير نوع الحقل الأخير

لانسى الفاصلة المنقوطة في نهاية السجل

التصريح عن متحول من نوع السجل (النوع الجديد):

اسم المتحول اسم السجل ;

• للوصول الى الحقل المطلوب في النوع المركب نضع :

; اسم الحقل نقطة اسم المتحول

البرنامج الثالث :

عرف سجل شخص يحوي اسمه ورقمه الوطني ، ثم اكتب برنامج يقوم بإدخال شخص وطباعة رقمه الوطني .

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

```
struct person {
```

```
char name [30];
```

```
long int id;
```

```
};
```

عرفنا السجل بإعطائه اسم **person** ويتالف من حقلين (**name**) للاسم و(**id**) للرقم الوطني وهو من النوع (**long int**) لأن الرقم كبير

❖ يمكننا التعريف عن الاسم باستخدام (**string name ;**) هنا استخدمنا النوع (**char**) وحجزنا 30 حرفاً من الذاكرة لاسم العامل بالشكل (**char name [30];**)

```
int main (){
```

```
person p ;
```

التصريح عن متغير من النوع الجديد

```
cin >> p.name >> p.id;
```

```
cout << "name is:" << p.name << endl << "number id is:" << p.id;
```

طباعة الاسم والرقم الوطني

```
return 0 ;
```

```
}
```

name is: Ali
number id is: 001256890

خرج التنفيذ

نتيجة التنفيذ بالشكل:

على فرض الاسم Ali

انتهت المحاضرة

إعداد دعاء الرحيل & سندس درويش & راما جوهر