◄ دكتور المادة: سمير جعفر

◄المحاضرة: الأولى ◄ عنوان المحاضرة: مراجعة لمفهوم الدوال

المحتوى العلمي : أهلاً بكم أصدقائي سندرس في هذه المحاضرة :

١-مراجعة لمفهوم الدوال

٢-تطبيق برنامجين باستخدام الدوال.

٣- بعض الملاحظات الهامة.

البنية الأساسية للدالة:

تعريف الدالة: برنامج جزئي يقوم بعمل محدد ، والبرنامج بلغة C++ يتألف من عدة دوال.

```
return – type    function name (list of parameters)
{
statement (1);
statement (2);
.
.
statement(n);
return value;
}
```

نوع الإرجاع — return – type

اسم الدالة — function name

قائمة الوسطاء — (list of parameters)

قوس بداية الدالة — {

التعليمات التي ستقوم الدالة بها. — statement (1); … statement(n);

ارجاع القيمة — return value;

قوس نهاية الدالة — }

شروط تسمية الدوال والمتغيرات:

- لا يبدأ برقم مثل: $3A$
- لا يحوي فراغات أو رموز خاصة باللغة.
- 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 الرئيسية .

التصريح عن دالة :

بذكر نوع ارجاعها واسمها مع قائمة الوسطاء.

مثلاً $int\ f\ (int\ x);$ وهذا ما ندعوه 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 مايتم 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 $(return\ ;)$.

فوائد استخدام الدوال:

- تفيد في تقليل السطور البرمجية
- تمنع التكرار لتعليمات معينة وسهولة قراءة البرنامج

البرنامج الأول :

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

الحل:

```
#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;}
```

يمكن كتابة اسم المكتبة بالشكل $\#include < iostream.h >$

لكن دون كتابة السطر $Using\ name\ space\ std;$

قمنا بتعريف دالة الطباعة والتصريح عنها قبل الدالة الرئيسة، لم نضع لها $return$ لأن نوع إرجاع الدالة $void$ ،

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.

نتيجة التنفيذ تكون بالشكل :

على فرض $n = 6$

نتيجة التنفيذ

$n = 6$

*** 6 ***

البرنامج الثاني :

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

١-حساب مضروب العدد n(العاملي $n!$).

٢-حساب قواسم العدد $n >$

٣-اختبار فيما إذا كان العدد أولي أم لا .

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

١-طباعة مضروب العدد n(عاملي n).

٢-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 << " "  ;
```

قمنا بتعريف دالة العاملي نوع إرجاعها النوع الصحيح لأن العاملي دائماً عدد صحيح أسمينا الدالة $fact$ والدالة تحوي وسيط واحد لتحسب العاملي له وترجع قيمته للدالة

قمنا بتعريف دالة من النوع $void$ لا ترجع قيمة هي فقط تطبع قواسم العدد

```
}

int  prim (int n ){
int t = 1
for (int i = 2; i < n; i + +)
if (n%i == 0)
{
t = 0
break;}
return t;}
int main( ){
int n ;
do{cout << "n = "; cin >> n;}
while (n < = 0);
cout << "n! = " << fact (n);
cout << endl;
int s = prim(n);
if (s == 1)cout << n << "is prim";
else div(n);
return 0 ;
}
```

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 باقي الاعداد فمجرد أن وجد عدد يقبل القسمة عليه دون باقي $(n\%i == 0)$ تخرج من الحلقة عن طريق $break$ وتعيد قيمة الاختبار.

بداية البرنامج

تصريح عن متحول نوع صحيح

قمنا باستخدام حلقة (do-while)لتقوم بإعادة إدخال العدد طالما انه سالب أو يساوي الصفر هذا الشرط لضمان العدد المدخل عدد موجب تماما

استدعينا دالة العاملي عن طريق تعليمة الطباعة مباشرة

نزول سطر

استدعينا دالة اختبار العدد الأولي عن طريق اسنادها لمتغير والمتغير نوع إرجاعه نفس نوع إرجاع الدالة

إذا كان العدد المدخل أوليا اطبع أنه أولي وإلا استدعي دالة divلطباعة القواسم

طريقة ثانية لاختبار فيما إذا العدد أولي ام لا:

```
bool prim(int n)
{bool t = true;
for (int i = 2; i < n; i++)
if (n%i == 0)
{t = false;
break;}
return t;}
```

فقط قمنا بتعريف دالة من النوع البولياني وعرفنا متحول من النوع البولياني نفترض في البداية أنه عدد أولي ***true*** ونختبر باقي الأرقام فإذا وجدت قيمة تقبل القسمة علية دون باق فيتم الخروج من الحلقة عن طريق ***break*** وترجع قيمة الاختبار

نتيجة التنفيذ بالشكل :

على فرض $n = 6$

```
n = 6
n! = 720
1  2  3
```

انتهت المحاضرة

"تذكر المسافة التي قطعتها وليس المسافة المتبقية لك فأنت لم تصل الى المكان الذي ترغبه ولكنك لست ايضا في المكان الذي كنت فيه"

إعداد: بيان البوشي*علا الدلاتي*مرح غريب