

c programming

Written By The-oNe

Part 2

بسم الله الرحمن الرحيم

• الإتفاقية:

لقد كتبت هذا الملف لغرض تثقيف الشباب العربي في مجال علوم الحاسب الآلي. وهذا الملف مجاني للجميع ولا أريد من ورائه إلا شيء بسيط جداً وهو دعوة صالحة في ظهر الغيب لي ولجميع أخواننا المسلمين في أنحاء العالم. كما أرجوا أن لا يتم التعديل في هذا الملف وإنسابه إلى غيري لأنني قد تعبت فيه كثيراً. كما أنني أوؤكد أنني أرحب وبكل سعة صدر بالنقد البناء الذي يستفيد منه الجميع. كما أنه إذا وجدت عزيزي القارئ أي أخطاء في هذا الملف يرجى إخباري بها وسيتم وضع أسمك في هذا الملف مع التعديل كما أنني أرحب أن يشترك معي أي شخص لإضافة المزيد من الدروس لهذا الملف وسيتم وضع أسمه أيضاً وذلك حتى يكون هناك مرجع عربي للغة السي.

• المتغيرات :

المتغيرات هي عبارة عن حجز مكان في الذاكرة لكي يتم ملؤها بقيمة ما . ويوجد هناك العديد من أنواع المتغيرات منها:

• متغيرات عددية صحيحة (integer numbers):

وهذه المتغيرات يمكن أن تكون القيم المسندة إليها صحيحة فقط سواء كانت موجبة أو سالبة ونقصد بأعداد صحيحة أي لا تحتوي على فاصلة عشرية. ويتم الإعلان عن هذه المتغيرات في جسم البرنامج. والإعلان يكون بذكر أولاً نوع المتغير وهنا نريد حجز متغير عددي لذلك سوف تكون طريقة الحجز هي (int a;) حيث أن ال (int) هي للأخبار أن هذا المتغير هو عبارة عن متغير عددي. أما ال (a) فهو اسم المتغير. ولا بد من ذكر الفاصلة المنقوطة كما قلنا سابقاً. ويمكن أن يتم الإعلان عن أكثر من متغير في سطر واحد مثل (int a,b,c;) وهكذا. كما ويفضل أن يتم الإعلان عن جميع المتغيرات من النوع الواحد في سطر واحد. لاحظ المثال التالي يبين كيف يتم التعامل مع المتغيرات من النوع العددي الصحيح:

```
#include <stdio.h>
main ()
{
int a,b;
a=5;
b=2;
printf("%d \n",a);
printf("%d \n",b);
return 0;
```

}

لاحظ المثال السابق يوجد به شيئين جديدين لم يتم التطرق إليها من قبل وهي أولاً (a=5) و (b=2) وهذي تعرف بعلمات الإسناد أي إسناد هذه القيمة لهذا المتغير وسيتم مناقشتها بإذن الله في قسم التعابير. كما قلنا سابقاً أن المتغير يشغل حيزاً من الذاكرة وهو يستخدم لتخزين القيم في المتغير. إذاً في حيز المتغير (a) ملأناه بالقيمة (5) وكذلك الحال مع المتغير (b). ويمكن أن يتم إسناد القيم عند التعريف بالصورة (int a=5,b=2;). أما الشيء الآخر الذي يعتبر جديداً هو (%d) وهذا يستخدم عند التعامل مع المتغيرات العددية الصحيحة وهو يعرف بموصفات المتغيرات وسيتم شرحها بالتفصيل في الفقرات القادمة. أي أن هذه العلامة ترمز دائماً أنه تم التعامل مع المتغيرات العددية الصحيحة. بعد تنفيذ المثال السابق سوف يتم طباعة التالي على الشاشة:

5

2

- متغيرات الأعداد الحقيقية الذي تحتوي على فاصلة عشرية (floating numbers):

وتكون القيم المسندة لهذا النوع من المتغيرات هي قيم عددية تحتوي على فاصلة عشرية. والأعداد الحقيقية هي الأعداد التي يكون فيها فاصلة عشرية سواءً كان العدد موجب أو سالب. وجميع ما ينطبق على المتغيرات العددية الصحيحة ينطبق أيضاً على هذا النوع. ويتم التعريف عن هذا النوع من المتغيرات على الصورة (float a) حيث كلمة (float) تعني أنه يجب أن يتم حجز مساحة من الذاكرة لمتغير من نوع حقيقي بفاصلة عشرية. أما (a) فهو اسم المتغير. والمثال التالي يبين كيفية التعامل مع هذا النوع من المتغيرات.

```
#include <stdio.h>
main ()
{
float a,b;
a=3.14;
b=8.1;
printf("%f \n",a);
printf("%f \n",b);
return 0;
}
```

في المثال السابق نلاحظ أنه يشبه المثال في المتغيرات العدية من النوع الصحيح ولكن الفرق هنا فقط في العلامة (%f) وهذه العلامة تعني أننا نتعامل مع المتغيرات من النوع العددي الحقيقي بالفاصلة العشرية. وسوف يكون خرج البرنامج السابق هو :

3.14

- المتغيرات الحرفية (character):

ويمكن إسناد قيم حرفية فقط لهذا النوع من المتغيرات حيث أن كل متغير يتسع فقط لتخزين حرف واحد فقط. ويمكن الإعلان عن هذه المتغيرات من خلال الصورة (char a;) حيث أن (char) هي لكي يتم إعلام جهاز الحاسب أن يحجز حيزاً من الذاكرة لمتغير من نوع حرفي. والمثال التالي يبين كيفية التعامل مع المتغيرات الحرفية:

```
#include <stdio.h>
main ()
{
char ch;
ch ='a';
Printf("%c \n",ch);
Return 0;
}
```

وهنا جميع ما ذكر في أمثلة المتغيرات السابقة فإنه ينطبق هنا في هذا المثال ما عدا أنه اختلف فقط العلامة (%c). وهي دلالة على استخدام المتغيرات الحرفية. ولاحظ أنه عند إسناد قيمة ما إلى متغير حرفي فإنه يجب أن تضع القيمة المسندة بين حاصرتي تنصيص (' ').
ويوجد أيضاً المتغير (double) للمتغيرات العددية الحقيقية ذات الفاصلة العشرية ويختلف هذا النوع عن ال (float) في أن هذا النوع يأخذ حجماً مضاعفاً من الذاكرة التي يأخذها المتغير (float).

- موصفات المتغيرات:

في أكثر البرامج الذي يتم كتابته يكون هناك تفاعل بين المستخدم لبرنامجك وبين البرنامج نفسه وذلك من خلال عمليات طباعة معلومات على وحدات الخرج الرئيسية لدى المستخدم مثلاً على شاشة المستخدم. كما أنه في بعض الأحيان لا بد من أخذ المعطيات من المستخدم حتى يتم عمل برنامجك، وإن كنت ستأخذ من المستخدم معطيات فإنه ولا بد أن تكون لهذه المعطيات حيز من الذاكرة حتى يتم تخزين هذه المعطيات وقد تعلمنا من قبل أن المتغيرات تقوم بحجز هذا المتغير ولكن السؤال هنا هو كيف يمكن لمفسر السي أن يعلم أن المعطى يتم تخزينه في حيز متغير عددي صحيح أو متغير عددي حقيقي بفاصلة عشرية أو حتى متغير حرفي ، لذلك وجد ما يسمى بموصفات المتغيرات التي تجعل المفسر يقوم بتمييز المدخلات إن كانت ستخزن في متغيرات عددية صحيحة أو متغيرات عددية حقيقية أو متغيرات حرفية . وهذا يعني أنه عندما نريد أن نطبع على شاشة المستخدم مثلاً متغير يملأ حيز من الذاكرة فإنه لا بد أن يتم الطريقة

التي سيتم من خلالها وصف نوع المتغير. والجدول التالي يبين الموصفات مع نوع المتغيرات المقترنة بها:

نوع المتغير المقترن بها	الموصف
هذا الموصف يصف المتغيرات العددية الصحيحة أي إذا أردنا طباعة محتوى متغير عددي صحيح على شاشة المستخدم فإنه يجب أن نستخدم هذه العلامة للدلالة على المتغير.	%d
هذا الموصف يصف المتغيرات العددية الحقيقية ذي الفاصلة العشرية ولا بد من استخدام هذه العلامة عندما نريد أن نعرف محتوى متغير عددي صحيح هذا المحتوى يكون مخزن في الذاكرة.	%f
وهذا الموصف يصف المتغيرات الحرفية ولا بد من استخدامه عندما نريد أن نعرف محتوى متغير حرفي هذا المحتوى يكون مخزن في الذاكرة.	%c
وهذا الموصف يستخدم للسلاسل الحرفية (مصغوفة من الأحرف) وسيتم مناقشتها لاحقاً.	%s

ويوجد أيضاً هناك العديد من هذه الموصفات ولكن هذه التي تهتمنا في دروسنا هذه. لاحظ المثال التالي كيف يمكننا من خلاله التحكم في طباعة المخرجات على شاشة المستخدم باستخدام الموصفات.

```
#include<stdio.h>
main ()
{
int a=65;
printf("%d\n",a);
printf("%c\n",a);
printf("%f\n",a);
return 0;
}
```

سيكون خرج هذا البرنامج هو :

```
65
A
Floating point not loaded
```

في السطر الأول من طبع العدد (65) وذلك لأن الموصف الذي استخدم هو موصف المتغير العددي الصحيح (%d). أما السطر الثاني فطبع الحرف (A) وذلك لأننا استخدمنا الموصف الحرفي (%c) وبما أن الحرف (A) يساوي العدد (65) في جدول الترميز أسكي إذن يتبين لنا أن لغة السي تستطيع التحويل وبكل بساطة بين الأنظمة العددية. أما السطر الثالث فمن المفترض أن يكون الخرج هو (65.00000) وذلك لأننا استخدمنا الموصف للمتغير العددي الحقيقي (%f) ولكن من أعتقد أنك هناك غلط ما من نفس المفسر والله العالم.

• الشروط اللازمة عند اختيار أسماء المتغيرات :

١. يمكن أن يكون من حرف واحد على الأقل أي الحروف من (a-z). ويمكن أن يكون اسم المتغير مكون من حروف وأرقام (0-9). مثل

```
Int abc;  
Float ab4;
```

٢. لا يمكن أن يبدأ اسم المتغير بحرف مثل :

```
Int 1gh;
```

لو حجزنا اسم المتغير بهذه الصورة فإن المفسر سوف يعطي رسالة خطأ.

٣. يمكن استخدام الشرطة التحتية (_) فقط في اسم المتغير مثل:

```
Int a_b;
```

٤. لا يمكن استخدام الشرطة (-) في اسم المتغير مثل:

```
Int a-b;
```

لو عرفنا المتغير بهذه الصورة فإن المفسر سوف يعطي رسالة خطأ.
٥. لا يمكن استخدام الفراغات في أسماء المتغيرات وإن حدث ذلك فإن المفسر سوف يعطي رسالة خطأ أيضاً.

```
Int as d;
```

٦. يمكن استخدام العدد الذي تريده من الأحرف في اختيار اسم المتغير.

٧. لغة السي حساسة بالنسبة لحالة الحروف أي أن الحرف (a) لا يساوي الحرف (A).

٨. لا يمكن أن يتم اختيار أحد كلمات اللغة المحجوز كأسماء متغيرات. مثل استخدام (main) كاسم متغير.

عند إختيارك لاسم المتغير فإنه من الأفضل أن يكون الاسم يدل على المتغير مثلاً لو كنت تريد كتابة برنامج لسرد أسماء الطلاب وأرقامهم فإن إختيار اسم المتغير (std_name) لأسماء الطلاب حيث أننا اختصرنا اسم المتغير من الجملة (student name)، أما متغير أرقام الطلاب فيكون اسم المتغير هو (std_num) وهو إختصار ل (student number).

وتقبلوا خالص تحيات أخوكم المحب (The-oNe)
الرجاء إرسال مقترحاتكم وآرائكم على العناوين التالية:

The-one@pharaonics.net

OR

The_o0ne@hotmail.com

OR

The_o0one@yahoo.com