



اسم المادة : برمجة الحاسوب  
اسم التدريسي : زيد ابراهيم رسول  
المرحلة : الثانية  
السنة الدراسية : 2023-2024  
عنوان المحاضرة: **FUNCTIONS**



**Ministry of Higher Education and Scientific Research**

**Al-Mustaqbal University**

**Computer Engineering Techniques Department**



**C/C++ programming language**

**Lecture (8)**

Prepared By

**MSc. Zaid Ibrahim Rasool**



اسم المادة : برمجة الحاسوب  
اسم التدريسي : زيد ابراهيم رسول  
المرحلة : الثانية  
السنة الدراسية : 2023-2024  
عنوان المحاضرة : FUNCTIONS



## FUNCTIONS

عملية تقسيم البرنامج ال دوال تعتبر من احدى المبادئ الرئيسية للبرامج المهيكل من خلال اتباع اسلوب من الاعلى الى الاسفل (Top Down), مما يؤدي الى امكانية استدعائها و استخدامها في اماكن مختلف من البرنامج.

### 5.1 فوائد الدوال

للدوال العديد من الفوائد يمكن تلخيصها كالتالي:

- 1- تساعد الدوال على توفير المساحات المطلوبة في الذاكرة.
- 2- تساعد الدوال المخزنة في ذاكرة الحاسوب او التي يكتبها المبرمج على تلافي عملية التكرار.
- 3- تساعد الدوال الجاهزة على تسهيل عملية البرمجة نفسها.
- 4- تقوم بعملية اختصار بزمان البرمجة و زمن تنفيذ البرنامج.
- 5- وجود امكانية استخدام الدوال مع برامج.
- 6- عندما يكون البرنامج مكون من دوال, فإن البرنامج نفسة يكون واضحا لدى المبرمج والقارئ والمستخدم على حد سواء.
- 7- تعمل الدوال على تجزئة البرنامج الطويل ال اجزاء صغيرة يمكن متابعتها واصلاحها من قبل المبرمج.

و لتوضيح اهمية الدوال في لغة ++C خذ المثال الاتي :

لو اردنا كتابة خوارزمية لخطوات صنع فنجان من القهوى فاننا نكتب مايلي:

- 1- ضع الماء في غلاية القهوى
- 2- سخن الماء حتى يغلي.
- 3- نضع القهوى في الماء.
- 4- نضف سكر اليه حسب الرغبة.
- 5- أطفئ النار.
- 6- صب القهوى في الفنجان.



اسم المادة : برمجة الحاسوب  
اسم التدريسي : زيد ابراهيم رسول  
المرحلة : الثانية  
السنة الدراسية : 2023-2024  
عنوان المحاضرة : **FUNCTIONS**



لنفرض الان اننا نود طلب فنجان قهوى من مقهى مجاور: فأن خطوات الخوارزمية التي نحتاجها الان هي خطوة واحدة فقط و هي:( استدعي فنجان قهوى).

تخيل الان كم وفرنا من الخطوات في حال لو استعملنا الدوال الجاهزة, بدلا من خطواتها التفصيلية و خاصة في برنامج يتطلب حسابات و عمليات كثيرة.

## 5.2 تعريف الدالة

تتكون الدالة من رأس و جسم, الصيغة العامة للدالة هي :

Type function\_name(argument1,argument2 ...)

Types of the parameter list variables;

```
{  
  
    types of local variables ;  
  
    function body ;  
  
    return (expression) ;  
  
}
```

**حيث** **Type** : يمثل نوع قيمة الدالة.

**function\_name**: يمثل اسم الدالة.

**argument1,argument 2**: تمثل دلائل او معاملات لاستقبال وارجاع البيانات.

**Types of the parameter list variables**: نوع المعاملات المستقبلية والمرجعة للبيانات.

**types of local variables**: المتغيرات المعلن عنها داخل الدالة ( وهي المتغيرات التي يعلن عنها وتستخدم

في حدود الدالة ولايمكن التعرف عليها في البرنامج الرئيسي او اي دالة اخرى حتى لو كانت تحمل نفس الاسم).

**function body**: يمثل جملة او مجموعة من الجمل.

**Return**: تمثل جملة اعادة قيمة التعبير (**expression**) حسب نوع الدالة.



اسم المادة : برمجة الحاسوب  
اسم التدريسي : زيد ابراهيم رسول  
المرحلة : الثانية  
السنة الدراسية : 2023-2024  
عنوان المحاضرة: **FUNCTIONS**



نوع الدالة هو يعتبر اي نوع من الانواع المعروفة في لغة C++ مثل (int, float, char...etc), عند استخدام الدوال يجب دائما ان يحدد نوع الدالة و هذا النوع يمثل نوع القيم التي ستعاد بواسطة الدالة, بحيث كل دالة تعيد قيمة تمثل نتيجة معالجة الايعازات في الدالة, وفي حال لم تتم اعادة اي قيمة من الدالة بعد انتهاء تنفيذ الدالة عندها سيكون النوع (void) و هو يعني لاشيئ, والنوع (void) يعني عدم حجز اي مكان للمتغير في الذاكرة.

يمثل اسم الدالة هو الاسم الذي تستدعي به الدالة, و بما ان اسم الدالة مسبق بنوع فهذا يعني ان اسم الدالة هو معرف او متغير.

تمثل المدخلات القيم التي ترسل الى الدالة من خارج الدالة لغرض معالجتها داخل الدالة, هذه المدخلات من الممكن ان تكون عبارة عن وسيط واحد او اكثر من وسيط و من الممكن ان لا يكون هنالك اي وسيط, و عندها تكون الاقواس اما خالية او نضع فيها كلمة (void).

يمثل جسم الدالة, و هو الذي يحتوي على الايعازات, او الشفرة اللازمة لانتاج العمل, الذي من اجله كتبت الدالة, و تكون هذه الشفرة محددة بين القوسين الذان يمثلان البداية والنهاية للبرنامج, و من الممكن ان تكون هذه الشفرة ايعا واحد او اكثر.

كل دالة يجب ان تستخدم على الاقل اثنين من هذه الاقواس ( الاقواس المتسوية) على الاقل تبادا بالقوس المفتوح ( { ) و تنتهي بالقوس المغلق ( } ).

بالعادة التنفيذ ينتهي ب عبارة الاعداء (return), اذ ستعيد او تسند القيمة الناتجة من تنفيذ الدالة الى اسم الدالة وهي تمثل المخرجات للدالة.



اسم المادة : برمجة الحاسوب  
اسم التدريسي : زيد ابراهيم رسول  
المرحلة : الثانية  
السنة الدراسية : 2023-2024  
عنوان المحاضرة: **FUNCTIONS**



مثال على ذلك:

float volume (float x , float y, float z)

نلاحظ هنا ان كل وسيط سيتم الاعلان عنه بشكل منفصل ولايجوز الدمج, فمثلا الاعلان التالي يعتبر غير

صحيح

float (int x , float y,z)

ملاحظات مهمة :

- 1- يتم الاعلان والتعريف عن الدالة في بداية البرنامج وقبل الدالة الرئيسية ( main ).
- 2- يتم استدعاء الدالة داخل البرنامج عن طريق اسمها فقط.
- 3- يجب الاعلان عن نوع الدالة اذا كانت ترجع بقيمة من النوع غير الصحيح (int), وفي حالة عدم الاعلان عنها قبل استدعائها تكون قيمتها int تلقائيا. مثال على ذلك:

float larger ( ).

int temp ( ).

temp ( ).

- 4- يكون للدالة مهمة معينة تؤديها بدون ارجاع قيمة عند انتهائها.

Void printmessage ( ) or Void printmessage (void)

### Examples with different return types

```
void add (int a,int b){  
    int z;  
    z=a+b;  
    Cout<<z;  
}
```



اسم المادة : برمجة الحاسوب  
اسم التدريسي : زيد ابراهيم رسول  
المرحلة : الثانية  
السنة الدراسية : 2023-2024  
عنوان المحاضرة: **FUNCTIONS**



```
int add (int a,int b){
    int z;
    z=a+b;
    Return z;
}

char check (int a, int b){
    if (a==b)
        return 'T';
    else
        return 'F';
}

float divide (float a,float b)
{
    float z;
    z=a/b;
    return z;
}
```



اسم المادة : برمجة الحاسوب  
اسم التدريسي : زيد ابراهيم رسول  
المرحلة : الثانية  
السنة الدراسية : 2023-2024  
عنوان المحاضرة : FUNCTIONS



**EX: Write a program in C++ to add these numbers**

1+5=          2+7=          5+4=

```
#include <iostream>
using namespace std;
void main (){
    int a,b,c;
    a=1,b=5;
    c=a+b;
    cout <<a<<"+"<<b<<"="<<c<<endl;
    int d,e,f;
    d=2,e=7;
    f=d+e;
    cout <<d<<"+"<<e<<"="<<f<<endl;
    int g,h,i;
    g=5,h=4;
    i=g+h;
    cout <<g<<"+"<<h<<"="<<i<<endl;
}
```



While when we use the function the program can be written as

```
#include <iostream>
using namespace std;
void add (int a,int b)
{
    int z;
    z=a+b;
    cout<<a<<"+"<<b<<"="<<z<<endl;
}
void main ()
{
    add(1,5);
    add(2,7);
    add(5,4);
}
```

By solving the same example with a function of return value

```
#include <iostream>
using namespace std;
int add (int a,int b)
{
    int z;
    z=a+b;
    return z;
}
```





اسم المادة : برمجة الحاسوب  
اسم التدريسي : زيد ابراهيم رسول  
المرحلة : الثانية  
السنة الدراسية : 2023-2024  
عنوان المحاضرة: **FUNCTIONS**



```
void main ()  
{  
    cout<< add(1,5);  
    cout<< add(2,7);  
    cout<< add(5,4);  
}
```