



الأسبوع "الثامن" برمجة حاسوب "C++"

اسم الطالبة: سندس عودة
اسم الدكتور: د. محمد خليل

Functions

Void → لا يرد قيمة
int, double, char, bool, ... → يرد قيمة



الكتابة بهذه الطريقة
في حال استدعاء دالة
لا يرد قيمة
void print (int) و ()

```
int main ( )  
{  
    ① cout << "x" << endl;  
    ② print ( ) ;  
    ③ cout << "A" << endl;  
    ④ cout << "B" << endl;  
    ⑤ cout << "C" << endl;  
}  
}
```

← استدعاء الدالة function
بعد تنفيذ الـ function يعود إلى مكان الاستدعاء

* كل function يقوم بحل مشكلة محددة

function :- a collection of statements to perform a task.

```

void info (int x)
{
    cout << x << endl;
    cout << x/2 << endl;
    cout << x*2 << endl;
    cout << x-1 << endl;
    cout << x+1 << endl;
}

```

```

int main ( )
{
    info(4)
    cout << " _____ \n" ;
    info(5)
    cout << " _____ \n" ;
    info(6)
    cout << " _____ \n" ;
}

```


Function definition

header of function & Function .

id ^{الاسم} _{function} [↑] name [↑] parameters

```
double abs (double number)
```

Function body

```
{
    if (number < 0)
        number = - number;
    return number;
}
```

لازم يكون نفس القيمة (نوع القيمة)
المكتوبة في ال header وهذا المقصود
بالنوع هو double

يعبر هذا ال Function قيمة واحدة وتكون من نوع double

header

* التي التي يميز كل Fun * يحبرني جانوبي البيانات

عن الآخر هو ال header المرحلة ونوع المخرجات من ال Function

Void Fun (int x)

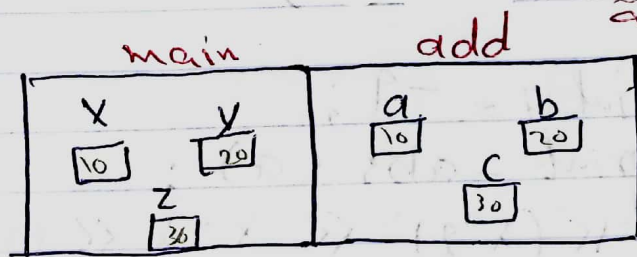
معنى كلمة Void هنا انه لا يعبر قيمة وتكون ال return
فيه اختيارية يتم اضافة بعض الاحيان لطباعة
عدد لمرة واحدة في ال main يكون * مقود * عن سطر لي لا يكون
ضمن امر طباعة او امر جمع او اعداد ... لانه لا يعبر قيمته
وتعمل في ال Void Fun عن شكل break.

Void print Name ()

```
{
    cout << "Omer" << endl;
    return ;
}
```

* لا تضع شي، لانه لا يعبر

يعود قيمة
 {
 integer
 int add (int a, int b) ③
 {
 int c = a + b; ④
 return c; ⑤
 }
 int main ()
 {
 ① int x = 10, y = 20;
 ② int z = add (x, y);
 ⑥ cout << z << endl;
 }



Out put → 30

مثال على طريقة عمل return في الـ Function
 void Fun (int x) ← local variable
 {
 ② cout << x << endl;
 ③ cout << x / 2 << endl;
 ④ return;
 cout << x - 2 << endl;
 cout << x + 1 << endl;
 }
 int main ()
 {
 ⑤ Fun (4); ①
 }

Out put

4
2

Function prototypes

~~we~~ place function definition before calling function's definition

```
① void print-abs (int number) } function
{                               } definition
    if (number < 0)
        number = - number;
    cout << number;
}

int main ( )
{
    int a = -9;
    print-abs (a);
    cout << "(x^2)" << x * x << endl;
}
```

ويمكن تعريف الـ Function بهذه الطريقة:

```
② prototype [ void print-abs (int number) ]
Function

int main ( )
{   int x = -9;
    print-abs (x);
    cout << "(x^2)" << x * x << endl;
}

[ تعريف ] void print-abs (int number)
Function {
    if (number < 0)
        number = - number;
    return number;
}
```

لا يمكن اجباري كتابة ثانياً والمختبرات
دنيا! اجباري كتابة النوعي والعدد

كتابة Function لإيجاد الـ (gcd) بين عددين

```
int gcd (int x, int y)
```

```
{  
    while (y != 0)
```

```
{  
        x = x % y;
```

```
        int temp = x;
```

```
        x = y;
```

```
        y = temp;
```

```
    }
```

```
    return x;
```

```
}
```

```
int main ( )
```

```
{
```

```
    int x, y;
```

```
    cin >> x >> y;
```

```
    cout << "gcd " << "x" << ", " << "y" << " " << "=" << " gcd(x,y)"  
    << endl;
```

```
}
```

كتابة Function لإيجاد عوامل العدد :-

```
void fun (int x)
```

```
{  
    for (int i = 1; i <= x; i++)
```

```
    {  
        if (x % i == 0)
```

```
            cout << i << " \t" ;
```

```
    }
```

```
void main ( )
```

```
{  
    int x;    cin >> x;
```

```
    fun (x);
```

```
}
```



```

int fun (int x);
int main ( )
{

```

① int x = 10;

② x = fun (x) + 5;

③ cout << fun(x) << endl;

}

```

int fun (int x)
{

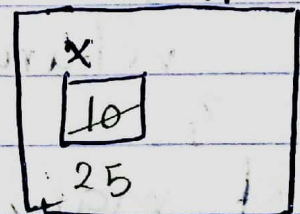
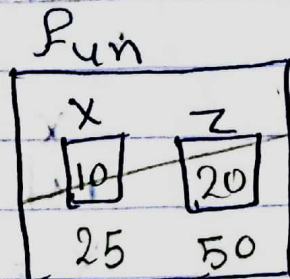
```

④ int z = x + x;

⑤ return z;

cout << z << endl;

}



Output = 50

The return statement

وبت برقع القيمة ① على مكان الاستدعاء

1) In an assignment statement

`int z = sum(x, y);` تعيين قيمة لمُتغيّر جديد

2) In an output statement.

`cout << sum(x, y);` إخراج طباعة

3) In an expression.

`int z = 100 + sum(x, y);`

أكتب function يقوم بطباعة المجموع (digits of num)

```
int sumofdigit (int x)
```

```
{
```

```
    int sum = 0;
```

```
    while (x)
```

```
    {
```

```
        sum += x % 10;
```

```
        x /= 10;
```

```
    }
```

```
    return sum;
```

u) As a parameter in a function call.

```
int largest (int x, int y)
```

```
{
```

```
    if (x > y)
```

```
        return x;
```

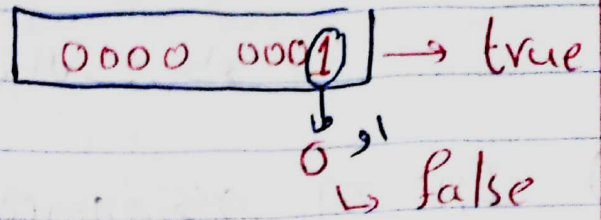
```
    }
```

```
void main () {
```

```
    int max = largest(a, largest(b, c));
```

bool

في الذاكرة
→



```
bool prim (int x)
```

```
{
```

```
    for (int i = 2; i <= x/2; i++)
```

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

```
            return false;
```

```
    return true;
```

```
}
```

```
void main ()
```

```
{
```

```
    for (int x = 10; x < 50; x++)
```

```
        if (prim(x))
```

```
            cout << "prime" << x << endl;
```

```
}
```


local and static variables.

Static `x = 5;`

يعرف مرة واحدة ويحفظ في الذاكرة حتى نهاية البرنامج

Global variables & Global Constant

local → داخل كود معين

* Const ثابت

* لا يمكن ابداء وتعريف عليه ان يكون بالحروف الكبيرة

Global تكون القيمة الابتدائية في حال عدم اعطائه قيمة هي الصفر

define YEAR 2020
لا يمكن التعديل عليها
Const int YEAR = 2020;
* الاختلاف في الاول كوضع (=) و () عند التعريف

define || long long
* في حال التعريف مرة أخرى في الـ #define يعطى error
اصلي طالع تعريفه مرة أخرى في const كـ (local) لا يصح مشكلة

write a C++ function that reverse an int number, and reverse true if the number is even otherwise false.

```
bool rev (int x)
{
    return (x%2)? false : true;
}

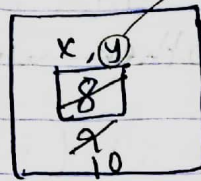
void main ()
{
    int x;
    cin >> x;
    cout << rev (x);
}
```

Function for checking if a number is even or odd. Yes if even, No if odd.

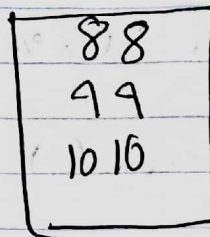
```
void fun (int x, int y)
{
    if ((x%2) == 0 && y%2 == 0)
    {
        cout << "yes" << endl;
    }
    else
    {
        cout << "No" << endl;
    }
}
```


Reference variables second name

```
int x = 8;
int &y = x;
```



```
cout << x << y;  
x++;  
cout << x << y;  
y++;  
cout << x << y;
```



output

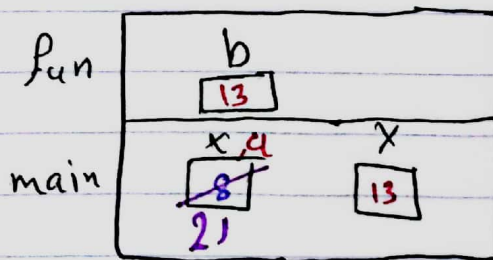
```
void fun (int &a, int b)
```

```
{  
    cout << "a = " << a << " b = " << b << endl;  
    a += b;  
}
```

```
int main()
```

```
{  
    int x = 8, y = 13;  
    cout << "x = " << x << " y = " << y << endl;  
    fun(x, y);  
    cout << "x = " << x << " y = " << y << endl;  
}
```

output



x = 8
a = 8
x = 21

y = 13
b = 13
y = 13

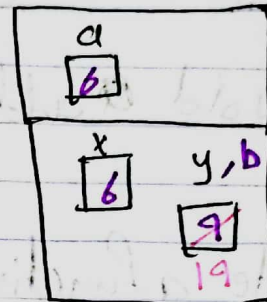
```
void addTen (int a , int b)
```

```
{
    a += 10;
    b += 10;
}
```

```
int main()
```

```
{
    int x = 6, y = 9;
    cout << x << y << endl;
    addTen (x, y);
    cout << x << y << endl;
}
```

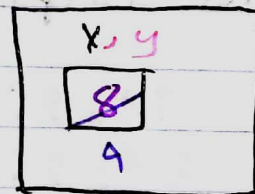
output → 6 9
6 19



add
Ten

main

```
int x = 8;
int y = x;
cout << x << y << endl;
x++;
cout << x << y << endl;
```



output → 8 8
9 9

```
y++;
cout << x << y << endl;
```

output
10 10

الطريقة الثانية إعلان عن وجود function يجب في هذه الحالة الكتابة داخل الـ **protocol** نوع البرمجة **ref**

```
void addTen (int, int);
```

```
int main { --- }
```

```
void addTen { --- }
```

Write a function called "duplicate" that receives three integers and duplicates them.

Duplicate means multiply each by 2;

مضاعفة الـ function
ذلك زلجا ان الـ **ref**
توجد قيمته واحدة فقط

```
void duplicate (int &a, int &b, int &c)
```

```
{
    a *= 2;
```

```
    b *= 2;
```

```
    c *= 2;
```

```
}
```

```
int main ( )
```

```
{
```

```
    int x, y, c;
```

```
    cin >> x >> y >> c;
```

```
    cout << x << " " << y << " " << c << endl;
```

```
    duplicate (x, y, c);
```

```
    cout << x << " " << y << " " << c << endl;
```

```
}
```

input	4	5	6
output	4	5	6
	8	10	12