



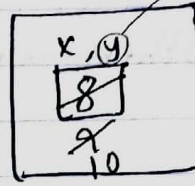
الأسبوع "العاشر" برمجة حاسوب "C++"

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

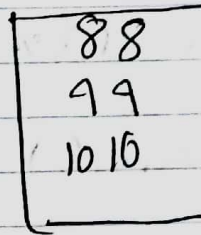
Reference variables

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)
```

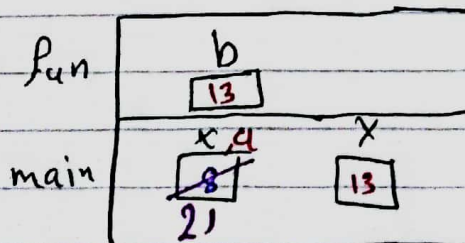
```
    cout << "x = " << x << " y = " << y << endl;
```

```
}
```

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;
```

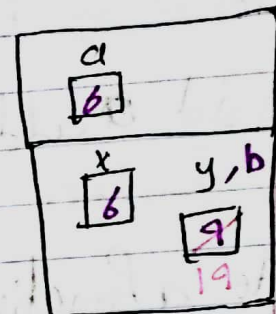
```
    cout << x << y << endl;
```

```
    addTen (x, y)
```

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

```
}
```

output → 6 9
6 19



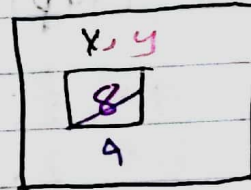
```
int x = 8;
```

```
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 انه

```
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;

منصة الـ Function
ذلك زلجاي الـ ref يرجع 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

Default Arguments

↳ **Default argument**: is an argument that is passed automatically to a parameter if the argument is missing on the function call.

```
int getsum (int x , int y = 6)
```

قيمة أولية في حال لم يتم إرسال
قيمة عند دالة الاستدعاء

```
return x + y ;
```

output

19
15

```
int main ( )
```

```
{
    int x = 9 , y = 10 ;
```

```
    cout << getsum ( x , y ) << endl ;
```

①

```
    cout << getsum ( x ) << endl ;
```

②

الرسالة

في الحالة رقم (1) عند استدعاء الدالة فإن القيمة

يعني

تقوم بإعداد القيمة الموجودة داخل تعريف الدالة

قيمة y في الاستدعاء الأول تكون القيمة المرسله وتكون y = 10

في الحالة رقم (2) حالة عدم إرسال قيمة للبرامتر الثاني " y " فإن النتيجة

بذلك

الموجودة في الـ header (نحل محلها وتكون

قيمة y تساوي 6

Main يجب وضع انه يوجد

* في حالة تعريف الدالة قبل

قيمة المتغير داخل الـ proto type

```
int getsum (int , int = 6)
```

* القيمة المرسله أقوى من القيمة الأصلية

```
void fun (int x = 40)
```

```
{
```

```
    cout << x;
```

```
}
```

```
int main
```

```
{
```

```
    fun();
```

```
    fun(6);
```

```
}
```

Output

```
40
6
```

إعطاء القيمة في حالة Default Argument تبدأ من اليمين ولا قبل وجود فراغات كإعطاء (int x = 5, int a, int y = 3) ① error

error (int x = 3, int a, int y)
يسار ماري

✓ (int x, int a, int y = 3)
من اليمين ولا يوجد فراغات

ولا يمكن حل هذه المشكلة (1) بهذه الطريقة مثلاً
int x = 3, int y = 4, int a = 5
cout << fun(x, y, a);
error

وأيضاً مشكلة (2) (int x = 3, int y = 4, int a = 5)
cout << fun(x, y, a);
error

الترك في هذه الحالة:

① يجب ألا نبدأ من اليمين أن إعطاء القيم تبدأ من اليمين
② لا يوجد فراغات

ومثلاً ككتابة كالمثال

في التعريف قبل الفونكشن prototype - من لازم اعطي اسم المتغير لكن
لازم اكتبه انه في قيمة ومن لازم اكتبها في header بعد كتابة
الفونكشن محامل


```
int get sum (int , int s 6);
```

```
int main ( )  
{
```

```
    int x = 9 , y = 10;  
    cout << getsum (x, y) << endl;  
    cout << getsum (x) << endl;  
}
```

```
int getsum (int x , int y)
```

```
{  
    return x + y;  
}
```

output

19
15

```
int sum (int a=0, int b=0, int c=0, int d=0)
```

```
{  
    return a+b+c+d;  
}
```

```
int main ( )
```

```
{
```

```
    int a=5, b=10, c=15 , d=20 ;
```

```
    cout << sum ( ) << endl;
```

```
    cout << sum (a) << endl;
```

```
    cout << sum (a, b) << endl;
```

```
    cout << sum (a, b, c) << endl;
```

```
    cout << sum (a, b, c, d) << endl;
```

```
}
```

output

0
5
15
30
50

Overloading Functions

Overloading Function : is ability to Create multiple function of the same name with different implementations.

كتابة أكثر من دالة بنفس الاسم لكن باختلاف عدد المتغيرات أو باختلاف نوع المتغيرات.

* يعني القيمة الواحدة (data type) أي في بداية الدالة ما إليها دخل ما يطلع عليها المتغيرات.

* ممنوع ترك خيار للمتغيرات عشوائي يوحّد قرار يعني الاستدعاء لازم يؤدي إلى دالة واحدة مش أكثر
يعني ممنوع Call يؤدي إلى أكثر من دالة

① باختلاف عدد المتغيرات

```
int sum (int a, int b)
```

```
{  
    cout << "Sum (int a, int b) = "  
    return a + b;  
}
```

```
int sum (int a, int b, int c)
```

```
{  
    cout << "Sum (int a, int b, int c) = "  
    return a + b + c;  
}
```

Output

```
Void main ( )
```

```
{  
    cout << sum (4, 5, 6) << endl;  
    cout << sum (3, 2) << endl;  
}
```

15

5

© باختلاف نوع datatype للبرامير

```
int sum (int a, int b)
```

```
{
    cout << "sum (int a, int b) = ";
    return a + b;
}
```

```
int sum (double a, double b)
```

```
{
    cout << "sum (double a, double b) = ";
    return a + b;
}
```

دالة أخرى

```
void getDimention (int, int, double, double)
void getDimention (int, int)
void getDimention (int, double)
void getDimention (double, double)
```

Call

```
getDimention (length)
getDimention (length, width)
getDimention (length, height)
getDimention (height, base)
```

int length, width
double base, height

(exact Matches)

يتم مقارنة اسم بين عملية الاستدعاء و (الفيكشن) exact


```

int Sum (int a, int b)
{
    cout << "Sum (int a, int b) = ";
    return a+b;
}

double Sum (double a, double b)
{
    cout << "Sum (double a, double b) = ";
    return
}

void main ()
{
    int x=5, y=10;
    double a=3.5, b=7.6;
    char ch1='A', ch2='B';
    cout << Sum (x, y) << endl;
    cout << Sum (a, b) << endl;
    cout << Sum (ch1, ch2) << endl;
}

```

Out put

```

Sum (int a, int b) = 17
Sum (double a, double b) = 11.1
Sum (int a, int b) = 131

```

بعض الاستدعاء في الحالتين الأولى والثانية exact matches
 بعض الاستدعاء في الحالة الأخيرة promotion
 جعلوا بروتوتيب في الحالة الأولى
 Char, short → الفرصتين وكان موصوف
 int → ما كان موجوداً في تعريف