



الأسبوع "الرابع" برمجة حاسوب "C++"

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

Control structures Repetition

For loops:

في For loop

هناك صيغة

رموز لتكتب

for (و ; و)

مثال 1 -

الطبوع الحرف "A" بعدة مرات

1) مرة واحدة 1)

cout << "A" << endl;

2) 1/2

for (و ; و)

{
cout << "A" << endl;
}

ب) لعدة مرات

for (int i = 0 ; i < 5 ; i++)
{
cout << "A" << endl;
}
خارج for

① T
②
③ condition
④
⑤ F

يقوم بعملية واحدة

out put

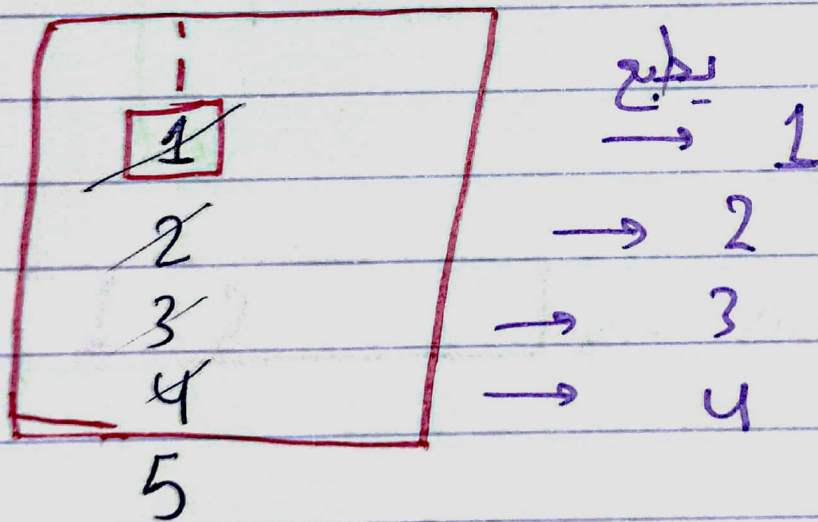
نقوم بتغيير الشرط حسب عدد المرات المطلوبة

i	يطبع
1	A
2	A
3	A
4	A
5	

2) مثال
for (int ^⑤ i = 1 ^① ; i < 5 ^③ ; i++)

② cout << i << " " ;

تكرر حتى يصبح جواب الشرط
 output اخرج



1 2 3 4

يطبع بهذا الشكل

```
for (int i = 1; i < 5; i++)
```

```
{
```

```
    cout << i; False
```

```
    if (i == 2) true
```

```
        break;
```

```
}
```

دائماً شرطیہ

i
1
2

1 2 ← out put.

Continue;

بہل کی چیز ہے
وہیج update

```
for (int i = 1; i < 4; i++)
```

```
{
```

```
    if (i % 2 == 0)
```

```
        'continue';
```

```
    else
```

```
        'cout << i';
```

```
}
```

i
1
2
3 4

1 3 ← out put

print all even numbers between 1 and 10 (inclusive)

print all even numbers between
1 and 10 inclusive

for (int i = 2; i <= 10; i = i + 2)

cout << i << endl;

2
4
6
8
10

2 4 6 8 10

← output

← output

endl; في حال

في حال "t"

نفس الفكرة للأعداد الفردية لكن العداد راح يبدأ بـ 1 بدل 2

program to print odd numbers
from 1 to 10

for (int i = 1; i < 10; i = i + 2)

cout << i << " \t " ;

i
1
3
5
7
9

1 3 5 7 9 ← output

الفرق بين main variables و local variables

Main Variables يبدأ من بداية البرنامج إلى نهايته

local Variables يتم تعريفه لمدة معينة وبعدها
تنتهي فترة حياته (من لحظة الإعلان عنه إلى حين نهاية blocks
المعلن عنه داخله)

```

1 void main ( )
2 {
3     for (int i = 0 ; i < 5 ; i++)
4         cout << i << " " << endl ;
5         cout << i ;
6     }

```

Output ← لن يخرج لنا output وسوف نظهر لنا هذه الرسالة

* error C2065: 'i': undeclared identifier

المتغير 'i' معرف فقط في المداخلة المحددة بين الأقواس الخاصة بـ `for` ولكن في حال عدم وجود أقواس خاصة بها فها يعتبر الـ compiler ان المتغير الذي يلي الـ `for` هو فقط تابع لها لذا يعتبر في المتغير 5 ان 'i' غير معرفة

وكذلك في المثال التالي لكن هنا بين لنا جملة `for` حيث حصلنا في اقواسها ()

```
void main ( )
```

```

{
    for (int i = 0 ; i < 5 ; i++)
    {
        cout << i << " " << endl ;
    }
    cout << i ;
}

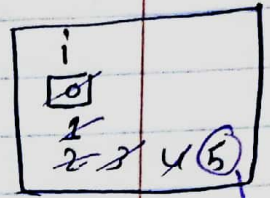
```

ومما سوف نظهر نفس الرسالة السابقة ✖

void main ()

int i;

for (i = 0; i < 5; i++)
 { cout << i << " \t" & i;
 cout << i;



نفسطيع استخدام المتغير (i) في هذا الطريقة لمرتكباً معرفاً عليه

يخرج من for بحال استخدام "i" هو من نقطة الاعلان حتى نقطة انهاء جملة او main (ربطها)

0 1 2 3 4 5 ← out put

for (int i = 0, j = 5; i < 5 && j > 2; i++, j++)
 cout << i * j << endl;

نفسطيع وضع أكثر من جملة داخل for

- ① initial
 - ② Condition
 - ③ update
- جميعها اختياري

يسمى Multiple Expressions

int i = 0;
 for (; i < 5; i++)
 cout << i << " \t" & i;

out put

0 1 2 3 4

أو
 int i = 0;
 for (; i < 5; i)
 { cout << i << " \t" & i;
 i++;

out put

0 1 2 3 4

في هذا النوع تكون شبيهة ب while

```

int i = 0;
for ( )
{
    cout << i;
    i++;
}

```

إذا كانت قيمة الشرط
غير موجودة هذا يعني أن جواب
الشرط دائماً **True** وقد ينتج
عنه **Infinite loop**

يمكن أيضاً باستخدام break (عالباً مُشروطة)

الـ **for loops** لها مخرجين هما **break** و **false** إذا لم يكن جواب الشرط

The syntax for a nested for loop statement in C++

```
for ( ① ; ② ; ③ )
{
```

- ① initial statement
- ② condition
- ③ update statement

```
    for ( ① ; ② ; ③ )
    {
        inner loop - statement(s) --;
    }
    outer loop - statement(s) ;
}
```

مثال :-

```
for (int i=0; i<3; i++)
{
    for (int j=0; j<4; j++)
    {
        cout << "Omar" << endl;
    }
}
```

تكرار (0, 1, 2)

i	j
0	0
	1
	2
	3

Omar تکرار
8 مرتبه

```
Omar
Omar
.
.
Omar
```

= Output

```
int x = 4;
for (int i = 0; i < x + 1; i++)
    cout << i << " " << x - i << endl;
```

x	i
4	0
4	1
4	2
4	3
4	4
4	5

→ False

```
for (int i = 0; i < 4; i++)
{
    for (int j = 0; j < 5; j++)
        cout << "A ";
}
```

output ← يفرض خطية A عشرين مرة

```
for (int x = 1; x < 11; x++)
{
    for (int i = 1; i < 11; i++)
        cout << i << " * " << x << " = " << endl;
```

i * x

output ← يفرض كتابة جدول الضرب لكل الأعداد المحصورة بين 1 و 10 (وأيضا أيضا)


```
int x = 5;
```

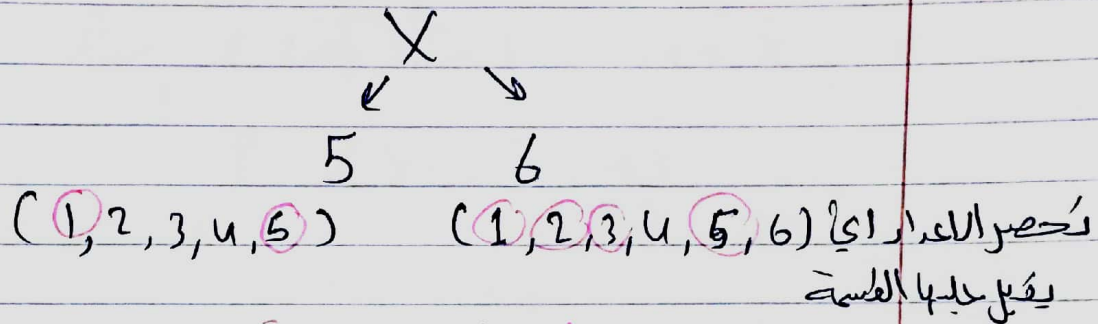
```
for (int i = 1; i <= 10; i++)
```

```
{  
    cout << i << "x" << x << " = " << i * x << endl;  
}
```

جدول الطرب لا 5 من 1 إلى 10

كتابة برنامج يظهر لنا ما إذا كان العدد الذي يدخله المستخدم
prime أو not prime

أولاً " نفرض أن المستخدم أدخل قيمتين أحدهما prime
والأخرى not prime



هذه القسمة

العدد الأولي لا يقبل القسمة إلا على عددتين
والباقى أي لا يقبل القسمة إلا على عددين

الحل :-

يتم يوم يحسب

عدد الأعداد

التي يقبل

العدد

المدخل

القسمة

عليها

```
int x, count = 0;
```

```
cin >> x;
```

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

```
{
```

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

```
{
```

```
count++;
```

```
}
```

```
}
```

```
if (count == 2)
```

```
{
```

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

```
}
```

```
else
```

```
cout << "not prime" << endl;
```

يقوم بفحص

إذا كان

عدد الأعداد التي

يقبل القسمة عليها

مساوية للعدد 2

فإن العدد prime

وإذا لا فإنه not prime