2022-2023

منصة تلاخيص منهاج أردني تقدم لكم

# تلخيص مادة الحاسوب

## الصف الحادي عشر - الفروع الأكاديمية والمهنية

الوحدة الثانية : البرمجة بلغة C++

إعداد وتصميم : أ. نعمة الأخرس


يمكنكم متابعة كل ما هو جديد والتواصل معنا من خلال :

0795360003 تلاخيص منهاج أردني YouTube تلاخيص منهاج أردني

# الوحدة الثانية: البرمجة بلغة (C++)

## الفصل الأول: مقدمة في لغة البرمجة c++

- .....:
- .....:

**البرمجة بلغة C++: هي إحدى لغات البرمجة الأكثر شيوعا وبدأ تطويرها كامتداد للغة C .**

**البرمجيات المستخدمة لكتابة برامج لغة C++:**

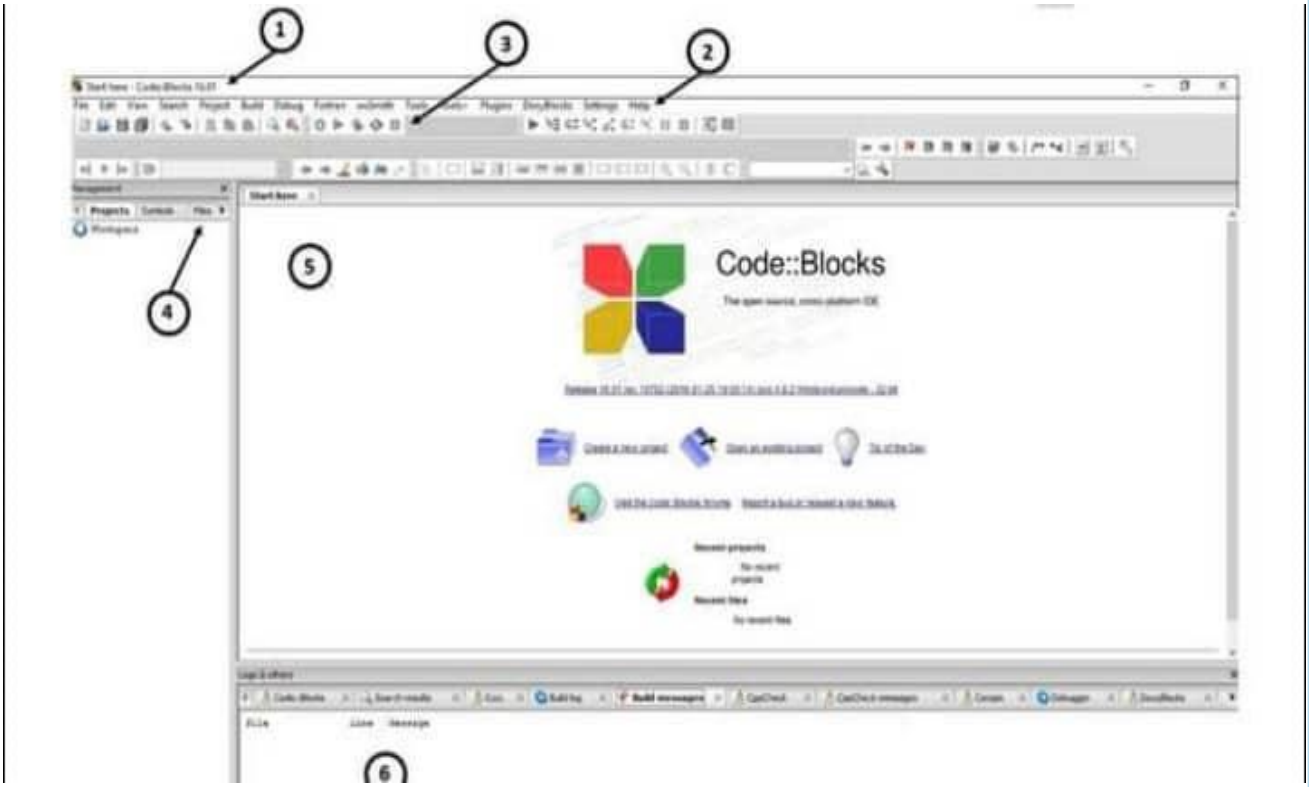
**1.Code lite**

**2.Code::BIocks**

**3.Visual Studio**

**اسم الموقع الذي يتم من خلاله تحميل برنامج Code Blocks: www.Codeblocks.Org**

**مكونات الشاشة الرئيسية Code Blocks:**



1. **شريط العنوان** : هو الشريط الذي يتضمن اسم البرمجية وعناصر التحكم بالنافذة من تصغير أو تكبير أو إغلاق .
2. **شريط اللوائح** : هو شريط يحتوي على مجموعة من اللوائح وكل لائحة تحتوي على مجموعة من الأوامر وكل أمر له وظيفة معينة .
3. **أشرطة الأدوات** : هي أشرطة تحتوي على مجموعة من الأدوات تؤدي كل منها وظيفة معينة ومن أهمها شريط Complier
4. **أدارة ملفات المشروع** : يقوم بعرض الملفات المتعلقة بالبرنامج الذي تعمل علية ويسهل التنقل بين الملفات .
5. **حيز العمل** : هو المكان الذي يظهر فيه أوامر البرنامج وجملة أثناء كتابة أو بعد استرجاعه
6. **منطقة الإعلام** : هي المنطقة التي تعرض الأخطاء التي وقعت فيها أثناء كتابة البرنامج

****خطوات إنشاء مشروع جديد :**

File ➔ New ➔ Project ➔ Console Application ➔ Go ➔ C++ ➔ Next ➔ ومن ثم
➔ تحديد اسم المشروع ومكان الحفظ ➔ Next ➔ Finish.

**Iostream: يقوم باستدعاء جمل الادخال Input وجمل الإخراج Output**

**int main () : هي الدالة الرئيسية المكونة لجميع برامج لغة C++ وهي أهم دالة في البرنامج وأساس البرنامج .**

**الشكل العام لأي برنامج بلغة C++**

```
#include <iostream>
using namespace std;
int main( )
{
 cout<<"Hello World!"<<endl;
 return 0;
}
```

**{: رمز بداية البرنامج**
**}: رمز نهاية أوامر البرنامج**
**return 0: نوع القيمة التي يرجعها البرنامج**

- **تنويه : أوامر لغة C++ يجب ان تكتب بالحروف الصغيرة ويجب أن تنتهي كل جملة بفاصلة منقوطة ;**

**بناء البرنامج Build: هي عملية التأكد من سلامة البرنامج من الأخطاء**

**طرق بناء البرنامج :**
1. **النقر على زر بناء البرنامج من شريط الأدوات Compiler**
2. **اختيار أمر Build من لائحة Build**
3. **الضغط على مفتاحي Ctrl + f 9**

**تنفيد البرنامج Run: هي عملية رؤية نتائج بعد التأكيد من خلوه من الأخطاء**

**طرق تنفيد البرنامج :**

1. **النقر على زر تنفيد البرنامج من شريط الأدوات Compiler**
2. **احتيار أمر Run من لائحة Build**
3. **الضغط على مفتاحي Ctrl = + f 10**

# الوحدة الثانية: البرمجة بلغة (C++)

## الفصل الثاني: الطباعة على شاشة المخرجات

**جملة الطباعة cout: تستخدم لطباعة الثوابت والمتغيرات وناتج العمليات الحسابية على شاشة المخرجات**

**الصيغة العامة لجملة الطباعة cout: cout<<data or variables; بحيث أن**

**cout: الأمر المستخدم لطباعة المخرجات على الشاشة وهي كلمة محجوزة .**
**<<: رمز يفصل بين أمر الطباعة cout وما يراد طباعته من بيانات ومتغيرات .**
**Data or variable: البيانات والمتغيرات التي سوف تطبع على الشاشة .**

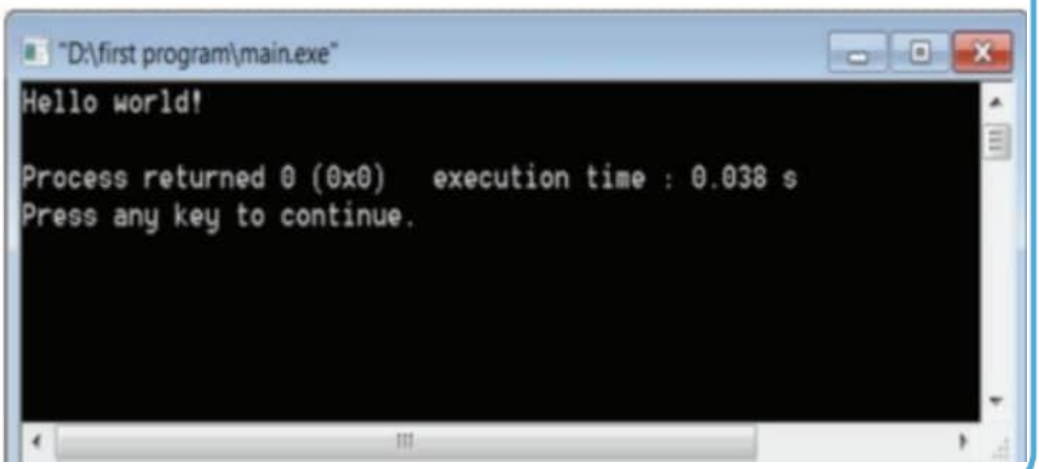
**ملاحظات هامة حول جملة الطباعة cout<< :**

- **تستخدم علامات التنصيص " " لطباعة الجمل النصية ولا تستخدم مع الأعداد الصحيحة أو الحقيقة .**
- **عند طباعة أكثر من قيمة فإنه يفصل بينها بالرمز <<**
- **يستخدم endl وهو اختصار لكلمتي end line لإنهاء سطر الطباعة والبدء من سطر جديد.**

**مثال 1 : البرنامج الآتي يطبع عبارة Hello World :**

```
#include <iostream>
using namespace std;
int main()
{
cout<<"Hello world!" <<endl;
return0;
}
```

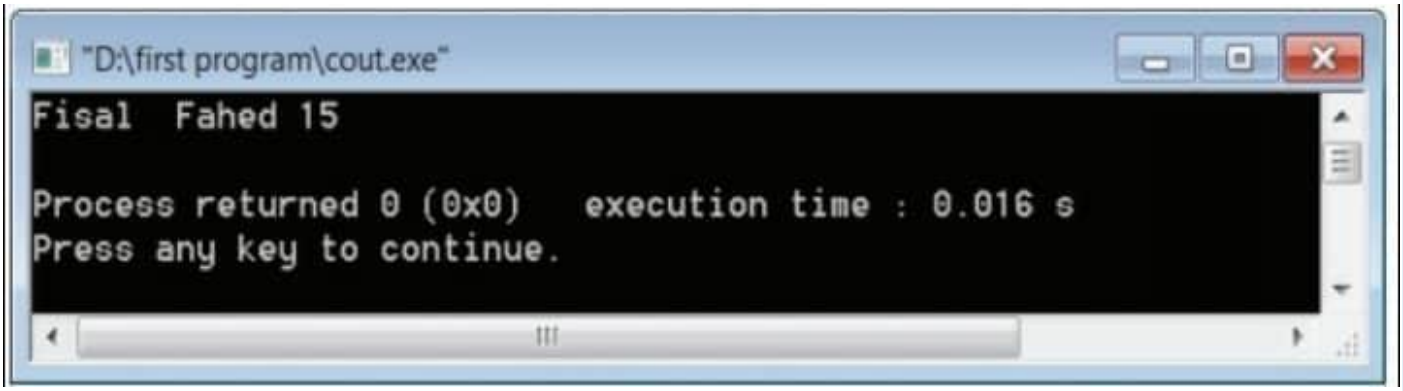
**شاشة المخرجات**



**مثال 2: البرنامج الآتي يطبع اسم Fisal Fahed و العدد (15) على نفس السطر :**

```
#include< iostream>
 using namespace std;
 int main ()
{
cout<<" Fisal Fahed "<<15<<endl;
return0;
}
```

**شاشة المخرجات**




**مثال3 : البرنامج الآتي يطبع اسم الطالبة Rana Ali على السطر الأول والمعدل 92.6 على السطر الثاني .**

```
#include<iostream>
using namespace std;
int main ( )
{
 cout<<"Rana Ali"<<endl;
 cout<<92.6<<endl;
 return 0;
}
```

**شاشة المخرجات**


**ملاحظة : يمكن اختصار جملتي الطباعة بجملة واحدة كما يلي :**

**cout<<Rana Ali<<endl<<92.6<<endl**

**سؤال : ما ناتج تنفيذ جمل الطباعة الآتية المكتوبة بلغة c++ كما تظهر على شاشة المخرجات ؟**

**1.cout<<"My name";**
**cout<<"is Reham";**

**……..**

**2.cout<<"My name"<<endl;**
**cout<<"is Reham"<<endl;**

**……**

**3.cout<<"My name"<<endl<<"is Reham";**

**……**

**4.cout<<"Average="<<87<<endl;**
**cout <<"Average=87";**

**……**

**رموز تنسيق مخرجات الجمل النصية Escape Sequence:**

**مكوناتها :**

- **الرمز\ ويسمى ب Backslash**
- **الأمر المراد تنفيذه**

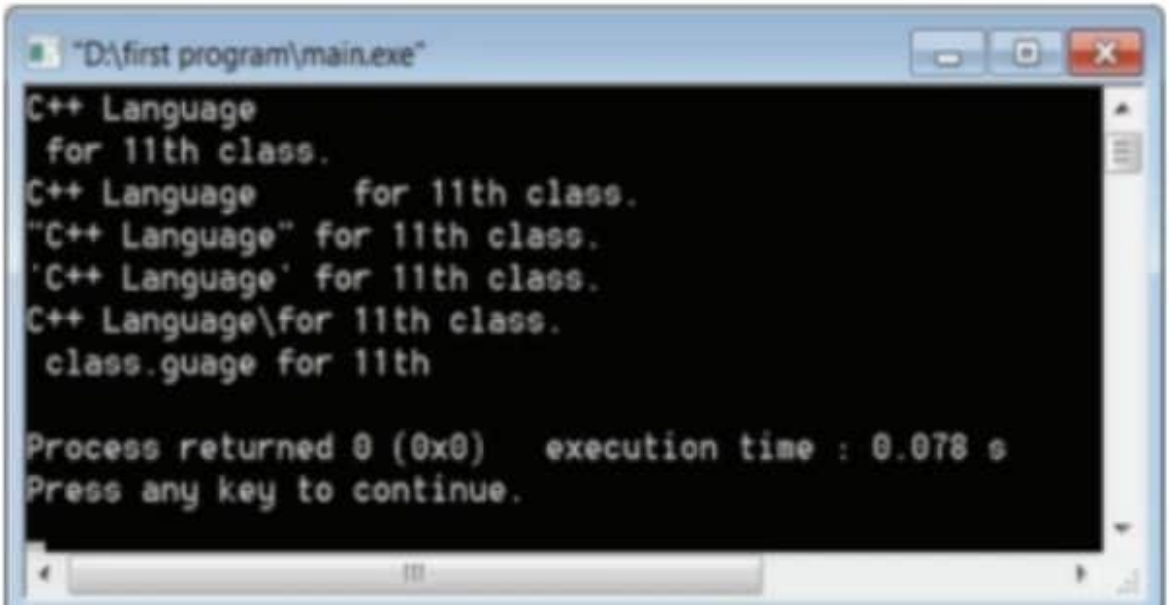
**تنبيه : يجب أن تكتب رموز مخرجات الجمل النصية بين " " فلا يجوز كتابتها خارج علامات التنصيص .**

| Escape Sequence | الوظيفة |
|---|---|
| \n | سطر جديد، ينتقل المؤشر إلى بداية سطر جديد. |
| \t | حقل، ينتقل المؤشر إلى الحقل الذي يليه. |
| \r | العودة للخلف، ينتقل المؤشر إلى بداية السطر الحالي. |
| \a | يصدر صوت الملاحظة الخاص بنظام التشغيل. |
| \\ | يطبع رمز (\) على الشاشة. |
| \' | يطبع رمز ( ' ) على الشاشة. |
| \" | يطبع رمز ( " ) على الشاشة. |

**مثال 1: ما ناتج تنفيذ جمل الطباعة الآتية المكتوبة بلغة c++ كما تظهر على شاشة المخرجات ؟**

```
#include <iostream>
using namespace std;
int main()
{
cout<< "C++Language \n for 11th class."<<endl;
cout<< "C++Language \t for 11th class."<<endl;
cout<< " \"C++Language \"for 11th class."<<endl;
cout<< "\' C++Language \' for 11th class."<<endl;
cout<< "C++Language \\ for 11th class."<<endl;
cout<< "C++Language  for 11th\r class."<<endl;
return 0;
}
```

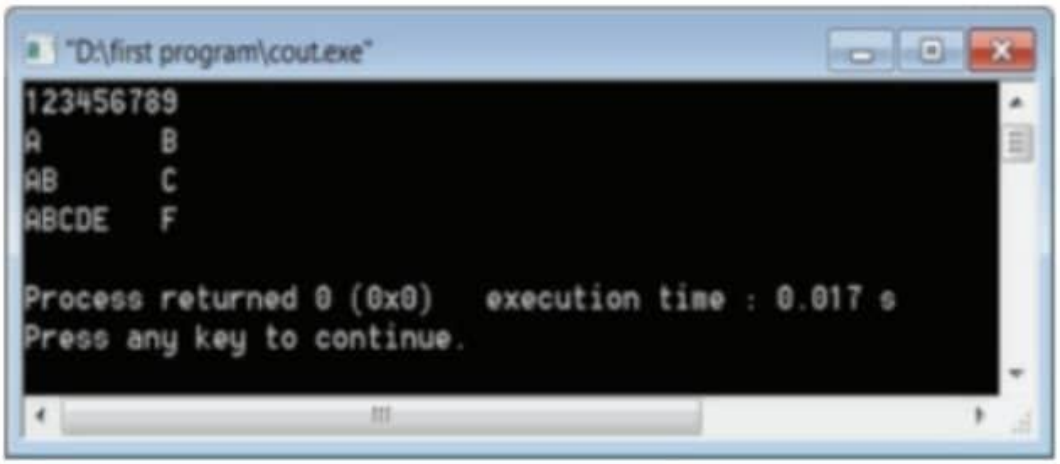
**شاشة المخرجات**




**مثال2 :**

```
#include <iostream>
using namespace std;
int main()
{
cout<< "123456789"<<endl;
 cout<< "A\tB"<<endl;
cout<< "AB\tC"<<endl;
cout<< "ABCDE\tF"<<endl;
return 0;
}
```

**شاشة المخرجات**




**إضافة الملاحظات بداخل البرنامج**

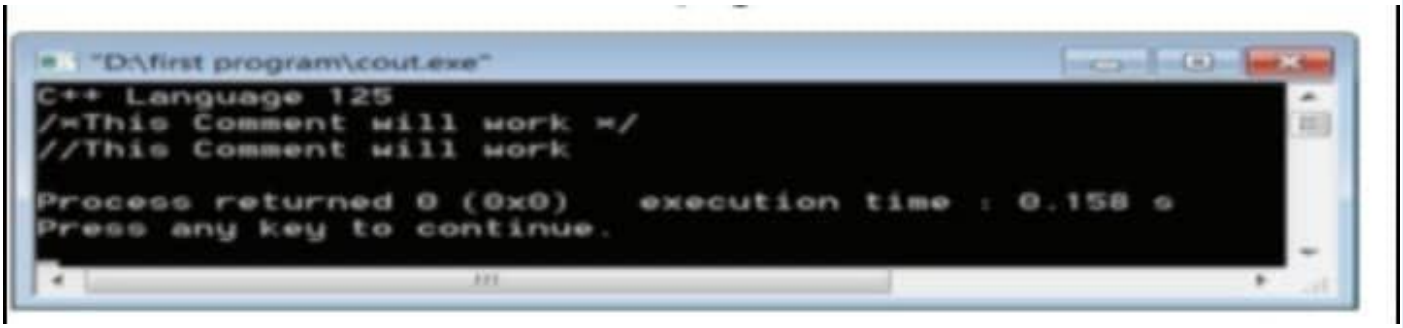
- **تستخدم إشارتي (\\) لإضافة ملاحظة إلى البرنامج**
- **اذا كانت أكثر من سطر فإنها تبدأ (/*) وتنتهي بإشارتي ( */)**

**ملاحظة : من الممكن أن تكتب الملاحظة في أي مكان في البرنامج وهي جملة غير تنفيذية أي لا يتم عند بناء البرنامج وتنفيذه ولا يوجد لها أي أثر على شاشة المخرجات .**

**مثال 1: ما ناتج تنفيذ جمل الطباعة الآتية المكتوبة بلغة c++ كما تظهر على شاشة المخرجات ؟**

```
/* ------------------------------------------------
This Program to Show the comments.
by 11th Computer Book Team
------------------------------------------------------*/
#include <iostream>
using namespace std;
int main()
{
//cout<< "Hello world!"<<endl;
cout<< "C++ Language ";//<<endl;
cout <<1 <<2  /* <<3 <<4 */ <<5<<  endl;
cout<< "/*  This comment will work  * / "<<endl;
cout<< "//  This comment will work   "<<endl;
return 0;
}
```

**شاشة المخرجات**



# الوحدة الثانية: البرمجة بلغة (C++)

## الفصل الثالث: المتغيرات وأنواع البيانات

**المتغيرات : هي أسماء تمثل مواقع في الذاكرة ذات قيم قابلة للتغير أثناء تنفيذ البرنامج ، ويعطي كل متغير اسم فريد ولا يتكرر ضمن البرامج و يحدد نوع المتغير حجم الذاكرة التي سوف تخصص له .**

1. **أسماء المتغيرات**

**هنالك عدة شروط يجب الالتزام بها عند اختيار أسماء المتغيرات وهي : -**

1. **يجب أن يبدأ اسم المتغير بحرف من حروف اللغة الإنجليزية( A-Z//a-z ) أو رمز الشرطة السفلية (-) Under score**
2. **أن لا يحتوي اسم المتغير على الرموز الخاصة والفراغات ،باستثناء الشرطة السفلية (-) وإشارة الدولار($)**
3. **أن لا يكون اسم المتغير من كلمات لغة c++ المحجوزة**
4. **يتكون اسم المتغير من أي عدد من حروف اللغة الإنجليزية أو رمز الشرطة السفلية أو الأرقام وفي حالة عدم الالتزام بتسمية المتغيرات بالطريقة السابقة ، سوف تظهر رسالة الخطأ ويتم تنفيذ البرنامج**

| اسم المتغير | سبب الخطأ في التسمية |
|---|---|
| 2Speed | لم يبدأ بحرف أو رمز (_) |
| Yahoo! | يحتوي على رمز (!) |
| #C++ | لم يبدأ بحرف أو رمز (_) ويحتوي على رمز (+) ورمز (#) |
| "Time" | لم يبدأ بحرف أو رمز (_) ويحتوي على رمز (") |
| A=B^5 | يحتوي على رمز (=) و (^) |
| – XY35@90 | يحتوي على رمز (@) |

**أنواع المتغيرات**

**تقسم المتغيرات حسب نوع البيانات المخزنة فيها**

**** أنواع البيانات ( Data Type) في لغة c++:-**

1. **الأعداد الصحيحة**
2. **الأعداد الكسرية**
3. **الرموز**
4. **الجمل النصية**
5. **القيم المنطقية**

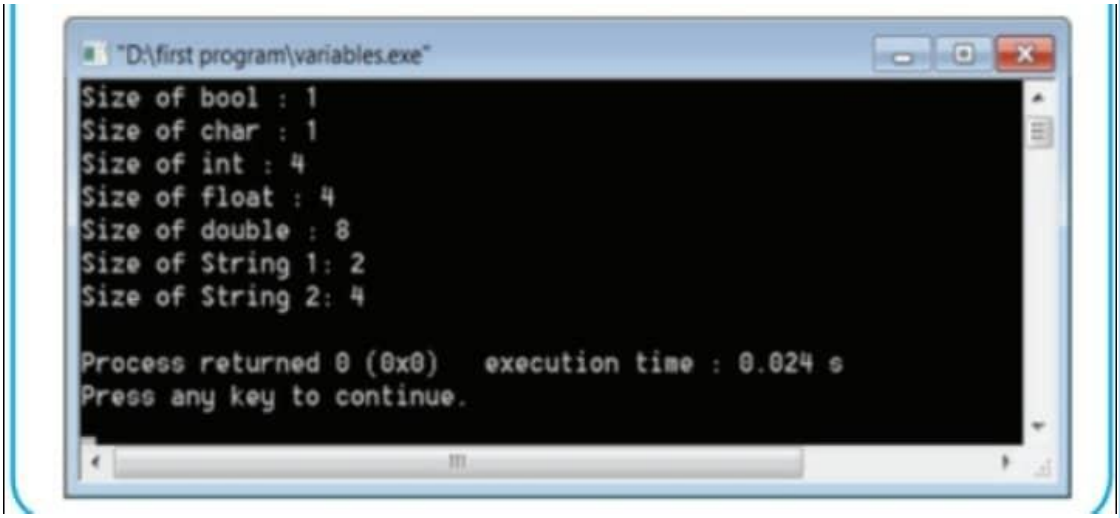
| النوع | الاستخدام |
|---|---|
| int | للأعداد الصحيحة. |
| float | للأعداد الكسرية. |
| double | للأعداد الكسرية الضخمة التي تتجاوز المليار. |
| char | لرموز اللغة المكونة من خانة واحدة. |
| string | للجمل النصية. |
| bool | للعمليات المنطقية (صح، خطأ). |

**علل: لماذا يتم حجز مواقع للمتغيرات في ذاكرة الحاسوب بالاعتماد على نوع البيانات المستخدمة للمتغيرات؟
ج: ذلك تقليل قدر الإمكان من حجز مواقع في الذاكرة دون الحاجة لها

1. نوع ( char ) يخزن رمزاً واحدا فقط لا يحتاج لأكثر من ( 1 byte )
2. الأعداد الصحيحة ( int ) فإنها تحتاج إلى ( 4 byte )

** size of():- يقوم حساب حجم نوع البيانات التي تم حجزها في الذاكرة وإظهار الناتج مباشرة

```
<include <iostream#
;using namespace std
()int main
{
cout<< "Size of bool : " <<sizeof(bool) <<endl;
cout<< "Size of char : " <<sizeof(char) <<endl;
cout<< "Size of int : " <<sizeof(int) <<endl;
cout<< "Size of float : " <<sizeof(float) <<endl;
cout<< "Size of double : " <<sizeof(double) <<endl;
cout<< "Size of String 1: " <<sizeof("A") <<endl;
cout<< "Size of String 2: " <<sizeof("H2O") <<endl;
   return0;
}
```




**ملاحظة :- حجم البايت الظاهر على شاشة المخرجات و يظهر الحجم يزداد بمقدار واحد على عدد الرموز ففي جملة" A" أظهرت المخرجات بأن الحجم بالبايت يساوي 2و1 . سبب لأن لغة c++ تقوم بإضافة رمز مخفي في نهاية كل جملة نصية يعبر عنها بكلمة ( NULL)لأن (NULL) تقوم بإعلام البرنامج بأن الجملة النصية قد انتهت**

**الكلمات المحجوزة : هي كلمات لها معنى خاص بالنسبة للحاسوب حيث يقوم الحاسوب بتنفيذ الأوامر المناظرة لمعاني هذه الكلمات .**
**ومن الأمثلة عليها في لغة c++:**

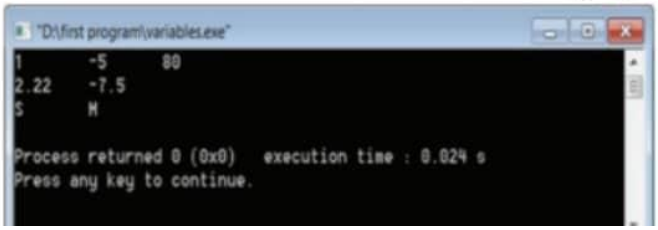
**يتم إسناد قيم المتغيرات بطريقتين : 1. الأولى بشكل مباشر أثناء كتابة البرنامج وتسمى إسناد قيمة للمتغير ( Assigned Values)**

**2. الثانية من خلال جمل الإدخال ( cin )**

**1. الإسناد بعد إنشاء المتغير :- تستخدم هذه الطريقة بعد إنشاء المتغير وفي أي مكان في البرنامج وتسمح لغة c++ بتغير قيمة المتغير في أي سطر من البرنامج وتستخدم هذه الطريقة بكثرة لتعديل قيم المتغيرات أثناء تطبيق البرنامج .**

```
#include <iostream>
using namespace std;
int main()
{
// Integer
    int i;
    int j;
    int k;
   // Float
   float f;
   float h;
   // Character
   char a;
   char b;
   // Assigned Value
   i = 1;
   j = -5;
   k = 80;
   f = 2.22;
   h = -7.5;
   a = ' S ';
   b = ' M ';
cout<<i<<"\t" << j << "\t" << k <<endl;
cout<< f << "\t" << h <<endl;
cout<< a << "\t" << b <<endl;
 return 0;
}
```

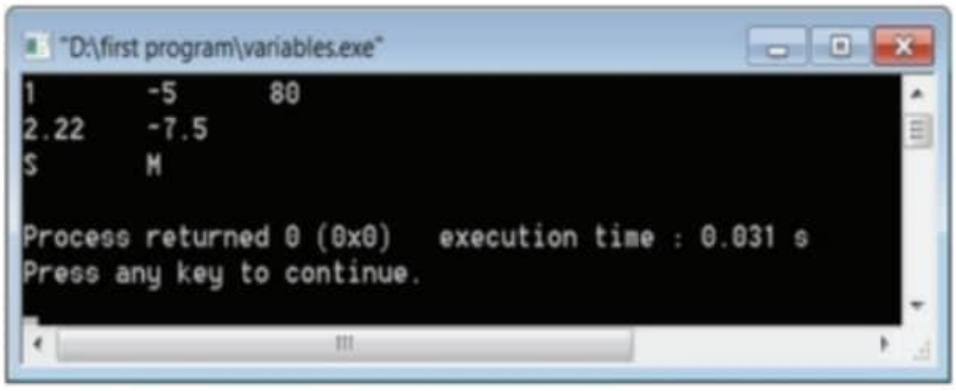
ناتج تنفيذ البرنامج:



**2. الإسناد عند الإنشاء :- تستخدم هذه الطريقة لإعطاء المتغير قيمة أولية عند إنشائه لأول مرة**

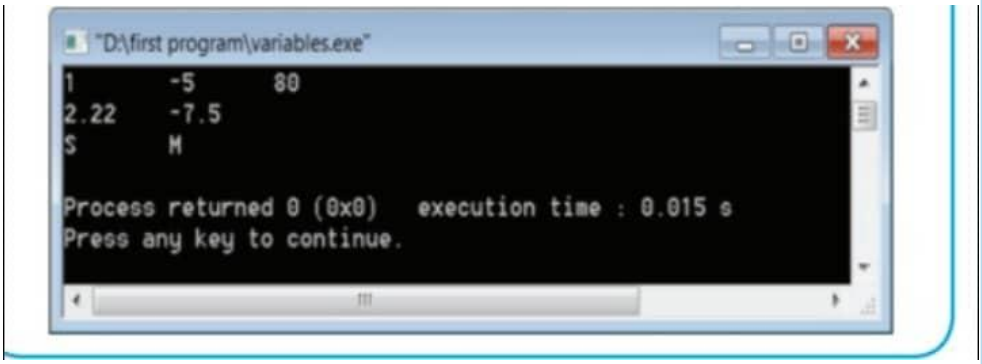
```
#include <iostream>
using namespace std;
int main()
{
// Integer
  int i = 1;
  int j = -5;
  int k = 80;
    // Float
    float f = 2.22;
    float h = -7.5;
    // Character
    char a = ' S ' ;
    char b = ' M ' ;
   cout<<i<< "\t" << j << "\t" << k <<endl;
   cout<< f << "\t" << h <<endl;
   cout<< a << "\t" << b <<endl;
    return 0;
}
```

ناتج تنفيذ البرنامج:



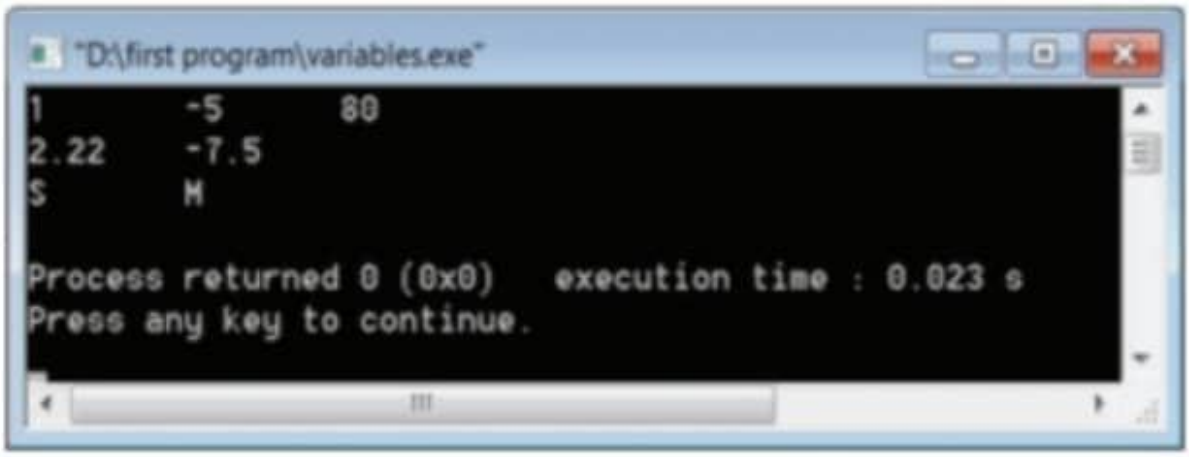
**** لتعريف عدة متغيرات من النوع نفسه يتم الفصل بين أسماء المتغيرات بفاصلة عادية( , )**

```
#include <iostream>
using namespace std;
int main ()
 {
  // Integer
  int i, j, k;
  // Float
   float f, h;
   // Character
   char a, b;
  // Assigned Value
  i = 1;
  j = -5;
  k = 80;
  f = 2.22;
  h = -7.5;
  a = 'S';
  b = 'M';
cout<<i<< "\t" << j << "\t" << k <<endl;
cout<< f << "\t" << h <<endl;
cout<< a << "\t" << b <<endl;
 return 0 ;
}
```



**** من الممكن اسناد قيم المتغيرات اثناء تعريفها مرة واحدة :-**

```
#include <iostream>
using namespace std;
int main ()
{
//Integer
int i = 1, j = -5, k = 80;
//Float
float f = 2.22, h = -7.5;
//Character
char a = 'S', b = 'M';
cout<<i<< "\t" << j << "\t" << k <<endl;
cout<< f << "\t" << h <<endl;
cout<< a << "\t" << b <<endl;
   return 0;
}
```



# الوحدة الثانية: البرمجة بلغة (C++)

## الفصل الرابع : جملة الإدخال cin

### أولا : التعريف بجملة الإدخال cin

**تستخدم جملة الإدخال cin: لإدخال قيم للمتغيرات في شاشة المخرجات عند تنفيذ البرنامج عن طريق لوحة المفاتيح**

- **الصيغة العامة لجملة الإدخال ( cin) هي :**

والصيغة العامة لجملة الإدخال (cin) هي:

```
cin>>variable-name;
```

حيث إن:

cin: الأمر المستخدم لإدخال البيانات إلى البرنامج عن طريق لوحة المفاتيح، وهي كلمة محجوزة.

>> : رمز الإدخال الذي يفصل بين المتغيرات.

variable-name: أسماء المتغيرات المطلوب إدخال قيم لها في شاشة المخرجات، ويجب أن تكون معرفة قبل جملة الإدخال في البرنامج.

**ملاحظات :**

1. **عند تنفيذ جملة الإدخال يتوقف البرنامج وتظهر إشارة مؤشر الفأرة على شاشة المخرجات التي تدل على انتظار إدخال البيانات**
2. **لا بد للمستخدم من إدخال قيمة أو أكثر حسب عدد المتغيرات الإدخال بينهما بفراغ ومن ثم الضغط على مفتاح الإدخال ( Enter)**
3. **مما يؤدي الى تخزين القيم المدخلة في المتغيرات الموجودة في جملة الإدخال**

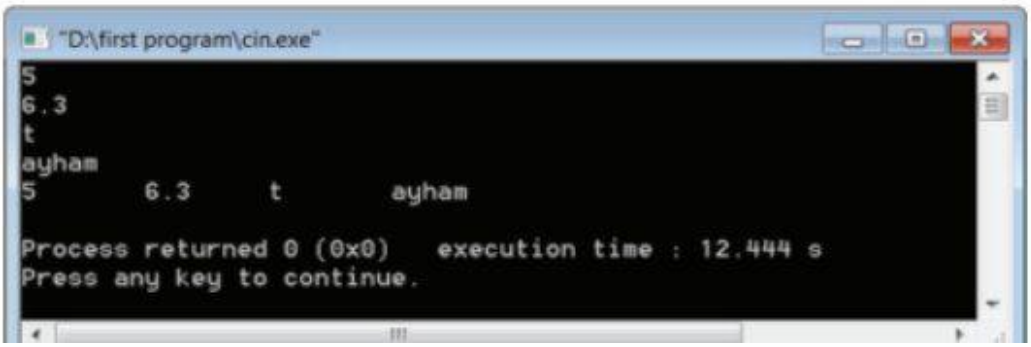
**مثال : يوضح كيف عملية إدخال قيم للمتغيرات المختلفة

يوضّح البرنامج الآتي عملية إدخال قيم للمتغيرات المختلفة.

```
#include <iostream>
using namespace std;
   int main ()
{
int a;
float b;
 char c;
 string d;
cin>> a;
cin>> b;
cin >>c;
cin>> d;
cout << a<<"\t"<<b<<"\t"<<c<<"\t"<<d<< endl;
return 0;
}
```

٩٧

ناتج التنفيذ هو:




**ثانيا : استخدام جملة الإدخال في كتابة برنامج C++**

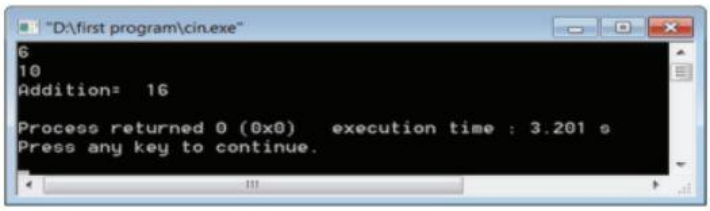
- **جملة الإدخال ( cin) المستخدم عند تنفيذها من تزويد الحاسوب بالبيانات اللازمة لإجراء عملية المعالجة المطلوبة**

****مثال : يوضح كيفية إدخال عددين صحيحين الى الحاسوب باستخدام جملة الإدخال cin تم إيجاد مجموعهما وطباعة الناتج**

```
#include <iostream>
using namespace std;
int main ()
{
int a,b,c;
cin>>a>>b;
c=a+b;
cout <<"Addition=\t"<<c <<endl;
return 0;
}
```

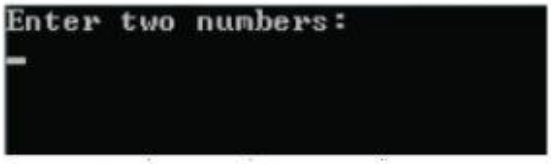
وعند تنفيذ البرنامج سيظهر مؤشر الفأرة على شاشة المخرجات منتظراً من المستخدم إدخال قيم للمتغيرات، ثم الضغط على مفتاح الإدخال(Enter)، بعدها سيقوم الحاسوب بتخزين القيم المدخلة في المتغيرات a،b، وإيجاد ناتج مجموعهما وطباعة الناتج، فيظهر الناتج كما يأتي:

٩٨




```
cout <<"Enter two numbers: " <<endl;
cin>>a>>b;
```

فعند تنفيذ البرنامج تظهر كالآتي:

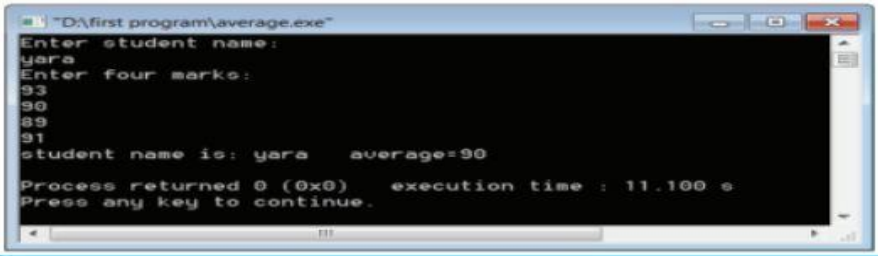



سيتوقف البرنامج منتظرا من المستخدم إدخال قيمتين عدديتين، وبهذا يتم إعلام المستخدم بالبيانات الواجب إدخالها.

**مثال : يوضح كيف الإدخال اسم طالب وأربع علامات مع توضيح طبيعة المدخلات ثم يحسب معدل الطالب ويطبع اسمه ومعدله**

```
#include <iostream>
using namespace std ;
int main()
{
string student_name ;
 int m1,m2,m3,m4;
 float average;
cout << "Enter student name: " <<endl;
cin >>student_name;
cout << "Enter four marks: " <<endl;
cin >>m1>>m2>>m3>>m4;
average=(m1+m2+m3+m4)/4;
cout << "student name is: " <<student_name<<" \t"<<"average=" <<average
<<endl;
return 0;
}
```

فيكون الناتج كما يأتي



# الوحدة الثانية: البرمجة بلغة (C++)

## الفصل الخامس : التعابير الحسابية Arithmatic Expressions

**أولا : ما المقصود بالتعابير الحسابية : بأنها عبارة عن قيم ثابتة أو متغيرات عددية أو مزيج من الثوابت والمتغيرات العددية يجمع بينهما عمليات حسابية**

**مهمة : تتشابه طريقة كتابة التعابير الحسابية في لغة C++ مع طريقة كتابة التعابير الحسابية الجبرية**

***يبين الجدول الآتي العمليات الحسابية في لغة C++:**

| العملية | المعنى | التعبير الحسابي الجبري | التعبير الحسابي في لغة C++ | مثال | ناتج المثال |
|---|---|---|---|---|---|
| + | الجمع | X+Y | X+Y | 6+2 | 8 |
| - | الطرح | X-Y | X-Y | 5-3 | 2 |
| * | الضرب | XY | X*Y | 5*4 | 20 |
| / | القسمة | $\frac{X}{Y}$ | X/Y | 16/8 | 2 |
| % | باقي القسمة | | X%Y | 14 % 3 | 2 |

**ثانيا: تنفيذ العمليات الحسابية**

١ – قواعد الأولوية لتنفيذ العمليات الحسابية

تستخدم لغة C++ قواعد الأولوية في تنفيذ العمليات الحسابية، وحسب التسلسل الآتي:

أ – في حالة وجود الأقواس ( )، يتم تنفيذ العمليات التي بداخلها أولاً.

ب– الضرب *، والقسمة /، وباقي القسمة.

جـ– الجمع +، والطرح –.

في حالة التكافؤ في الأولوية، يتم التنفيذ من اليسار إلى اليمين. والجدول (٢–٥) يبين كيفية كتابة بعض التعابير الحسابية بلغة C++.

الجدول (٢–٥ ):كتابة التعابير الحسابية بلغة C++.

| التعبير الحسابي | التعبير بلغة C++ |
|---|---|
| x+5y-xy | x+5*y-x*y |
| $y^2 - (x+3)$ | y*y-(x+3) |
| $\frac{x - y}{4y}$ | (x-y)/(4*y) |
| x(-3y+3) | x*(-3*y+3) |

**٢ – إيجاد ناتج التعبير الحسابي**

عند إيجاد قيمة تعبير حسابي في لغة C++، يجب مراعاة قواعد الأولوية والشكل (٢– ٢٠) يوضح مثالاً على تسلسل تنفيذ التعبير الحسابي وإيجاد نتيجته.

إذا كانت a=4،b=7،c=3 فإن ناتج التعبير الحسابي الآتي مع بيان تسلسل التنفيذ هو :

a – b % 2 * ( c+ 5)

4 – 7 % 2 * ( 3 + 5 )

4 – 7 % 2 * 8

4 – 1 * 8

4 – 8

-4

**ثالثا : معاملات الزيادة والنقصان القبلية والبعدية**

❖ **توفر لغة C++ معامل الزيادة أو Increment Operator ( ++) الذي يقوم بإضافة واحد الى قيمة المتغير ومعامل النقصان Decrement Operator (- -) الذي يقوم بإنقاص واحد من قيمة المتغير**

** **نوعان من معاملات الزيادة والنقصان هما ؟**

**١ – معامل زيادة أو نقصان قبلي (++x,--x)**

ونعني بالزيادة أو النقصان القبلي إتمام عملية الزيادة بمقدار واحد على قيمة المتغير أو النقصان بمقدار واحد من قيمة المتغير أولا، ومن ثَمَّ تنفيذ العملية المطلوبة سواءً كانت طباعة أو غيرها من العمليات.

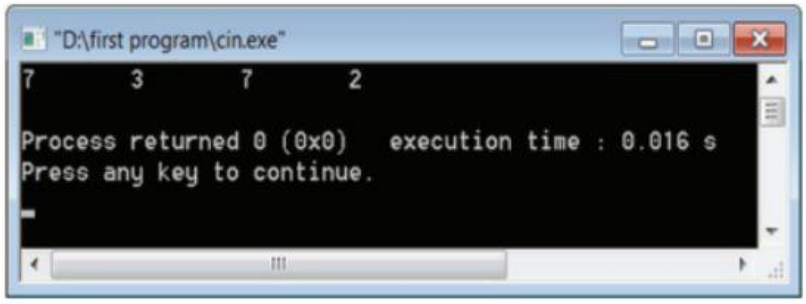
**٢ – معامل زيادة أو نقصان بعدي (x++, x--)**

ونعني بالزيادة أو النقصان البعدي إتمام العملية المطلوبة أولاً ثم إجراء الزيادة بمقدار واحد على قيمة المتغير أو النقصان بمقدار واحد من قيمة المتغير، ويوضح الشكل (٢– ٢١) مفهوم معامل الزيادة القبلية والبعدية.

عند تنفيذ البرنامج الآتي :

```
#include <iostream>
using namespace std;
int main()
{
   int x =6,y=2;
   int z,w;
   z=++x;   // x=7 , z=7
   w=y++;  //  w=2 ,y=3
   cout <<x <<"\t"<<y<<"\t"<<z<<"\t"<<w<<"\t" <<endl;
   return 0;
}
```

يظهر الناتج:




لاحظ عند تنفيذ الجملة z=++x; ، تمّت زيادة المتغير x بمقدار واحد، فأصبحت قيمته تساوي 7، ثم تمّ إسناد قيمة x للمتغير z فأصبحت قيمة z تساوي 7. وعند تنفيذ الجملة w=y++;، تمّ أولا إسناد قيمة المتغير y للمتغير w فأصبحت قيمة w تساوي 2 ، ثم تمّت زيادة المتغير y بمقدار واحد فأصبحت قيمة y تساوي 3.

❖ **قواعد الأولوية عند نتفيد العمليات الحسابية هما :**

1. **في حالة وجود الأقواس () يتم تنفيذ العمليات التي بداخلها أولا**
2. **معاملات الزيادة والنقصان القبلية**
3. **الضرب* ،والقسمة / ، وباقي القسمة**
4. **الجمع+ ، والطرح - .**
5. **معاملات الزيادة والنقصان البعدية**

**رابعا: معاملات الإسناد**

**هي معاملات تقوم بعملية حسابية مع إسناد قيم للمتغير في وقت واحد**

**يبين الجدول الآتي معاملات الإسناد في لغة C++

| معامل الإسناد | مثال | الوصف |
|---|---|---|
| + = | c+=2 | c=c+2 |
| - = | c-=5 | c=c-5 |
| * = | c*=8 | c=c*8 |
| / = | c/=5 | c=c/5 |
| % = | c%=3 | c=c%3 |

# الوحدة الثانية: البرمجة بلغة (C++)

## الفصل السادس : التعابير العلائقية والمنطقية

**أولا : ما المقصود بالتعبير العلائقي: هو جملة خبرية يكون ناتجها إما صوابا ( True) وإما خطأ ( False) ويكتب هذه التعابير باستخدام عمليات المقارنة**

**** تستخدم التعابير العلائقية : في جمل اتخاذ القرار ( جملة الشرطية - if ) في لغة C++**

**مهمة : تتشابه طريقة كتابة التعابير العلائقية مع طريقة كتابتها الجبرية**

الجدول (٢–٧): عمليات المقارنة.

| العملية | لفظ التعبير العلائقي | التعبير الجبري | التعبير بلغة C++ |
|---|---|---|---|
| > | X is greater than Y<br>x أكبر من Y | X > Y | X > Y |
| < | X is less than Y<br>x أصغر من Y | X < Y | X < Y |
| >= | X is greater than or equal to Y<br>x أكبر من أو تساوي Y | X ≥ Y | X >= Y |
| <= | X is less than or equal to Y<br>x أصغر من أو تساوي Y | X ≤ Y | X <= Y |
| == | X is equal to Y<br>X تساوي Y | X = Y | X == Y |
| != | X is not equal to Y<br>X لا تساوي Y | X ≠ Y | X != Y |

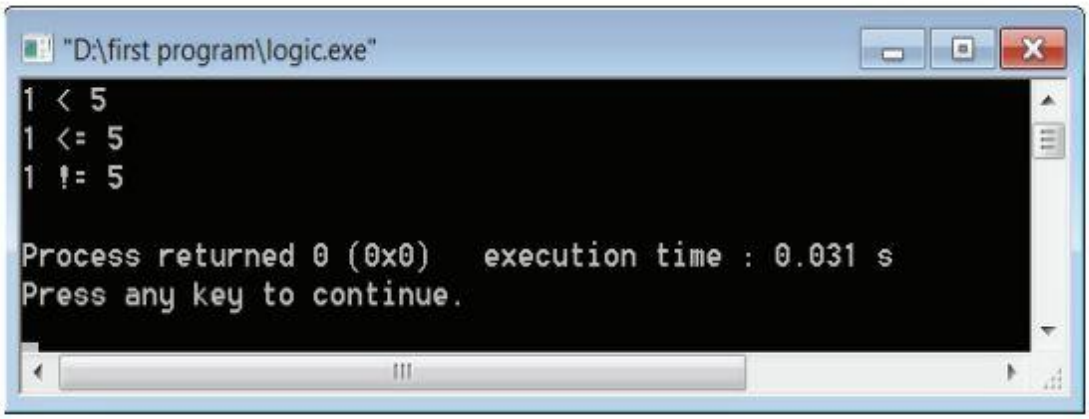
**** يبين الجدول الآتي بعض أمثلة على التعابير العلائقية وناتجها المنطقي ( صواب ، خطأ )**

| التعبير العلائقي | ناتج التعبير العلائقي |
|---|---|
| 2 + 5 == 8 – 1 | True |
| 5*2 != 10 | False |
| 7 % 3 >= 7 | False |
| 40 / 5 < 40 / 4 | True |
| 20 – 4 * 5 == 0 | True |
| 85<= 3 * 20 + 5 * 5 | True |

**** مثال : يوضح استخدام التعبير العلائقي في لغة C++**
****يكتب التعبير العلائقي بين قوسين بعد كلمة if واذا كان ناتج التعبير المنطقي صوابا تنفذ الجملة التي تليها واذا كان نلتجها خطأ سيتجاهل الجملة التي تليها**

```
#include <iostream>
using namespace std;
int main()
{
   int x = 1;
   int y = 5;
   if( x > y ) cout<< x << " > " << y << endl;
   if( x < y ) cout<< x << " < " << y << endl;
   if( x >= y ) cout<< x << " >= " << y << endl;
   if( x <= y ) cout<< x << " <= " << y << endl;
   if( x == y ) cout<< x << " == " << y << endl;
   if( x != y ) cout<< x << " != " << y << endl;
   return 0;
}
```

ناتج تنفيذ البرنامج:




**ثانيا : ما المقصود بالتعبير المنطقي : هو جملة خبرية تتكون علائقيين أو أكثر مرتبطة باستخدام المعاملات المنطقية ( And ,Or ) وتكون قيمتها إما صوابا وإما خطأ**

**** يستخدم المعامل المنطقي ( Not ) : لنفي التعابير العلائقية أو المنطقية**

❖ اذكر للمعاملات المنطقية هما ؟

1. معامل الربط ( AND) ويرمز له ( && )
2. معامل الربط ( OR) ويرمز له ( || )
3. معامل النفي (ويرمز له ( ! ) NOT)

## معامل الربط ( AND) ويرمز له ( && )

1. يستخدم المعامل AND: الربط التعابير العلائقية ويكون ناتج التعبير المنطقي صوابا ( True) إذا كان ناتج جميع التعابير العلائقية المرتبطة بها صوابا ( True)
2. ويكون ناتج التعبير المنطقي خطأ ( False) إذا كان ناتج أحدى التعابير العلائقية المرتبطة بها خطأ ( False)

** جدول الصواب والخطأ للمعامل ADN:

| A | B | A && B |
|---|---|---|
| FALSE | FALSE | FALSE |
| FALSE | TRUE | FALSE |
| TRUE | FALSE | FALSE |
| TRUE | TRUE | TRUE |

** الأمثلة الآتية توضح استخدام المعامل ( ADN) في التعابير المنطقية

مثال (١)

```
5 > 3 && 5 < 10
True && True
True
```

مثال (٢)

```
7 >= -8 && 2 <= 1
True && False
False
```

## معامل الربط (OR) ويرمز له ( || )

1. يستخدم معامل الربط (OR) : لربط التعابير العلائقية ويكون ناتج التعبير المنطقي صوابا ( True) إذا كان ناتج إحدى التعابير العلائقية المرتبطة بها صوابا ( True)
2. ويكون ناتج التعبير المنطقي خطأ ( False) إذا كان ناتج كل التعابير العلائقية المرتبطة بها خطأ ( False)

** جدول الصواب والخطأ للمعامل ( OR ) :

| A | B | A \|\| B |
|---|---|---|
| FALSE | FALSE | FALSE |
| FALSE | TRUE | TRUE |
| TRUE | FALSE | TRUE |
| TRUE | TRUE | TRUE |

** الامثلة الآتية توضح استخدام المعامل OR في التعابير المنطقية :-

مثال (١)

```
5 > 3 || 5 < 10
True || True
True
```

مثال (٢)

```
7 >= -8 || 2 <= 1
True || False
True
```

معامل النفي ( NOT ) ويرمز له ( ! )

1. يستخدم معامل النفي ( NOT ) : لنفي التعابير العلائقي أو المنطقي فإذا كان ناتج التعبير صوابا ( True ) فإنه ينفيه ليصبح خطأ ( False )
2. وإذا كان ناتج التعبير خطأ ( False ) فإنه ينفيه ليصبح صوابا ( True )

** جدول الصواب والخطأ للمعامل NOT

| A | !A |
|---|---|
| FALSE | TRUE |
| TRUE | FALSE |

** الأمثلة الآتية توضح استخدام TON في التعابير العلائقية والمنطقية :

مثال (١)

```
! ( 7 > -1)
! ( True)
False
```

مثال (٢)

```
! ( 5 > 3  &&  5 < 10 )
! (True  &&  True)
! (True)
False
```

مثال (٣)

```
! ( 7 >= -8  &&  2 <= 1 )
! ( True  &&  False )
! ( False)
True
```

تستخدم التعابير المنطقية أكثر من معامل لربط التعابير العلائقية إذا كانت هذه التعابير العلائقية ثلاثة تعابير أو أكثر، وفي هذه الحالة يجب تطبيق قواعد الأولوية لتنفيذ التعابير المنطقية، حسب التسلسل الآتي:

١ – الأقواس

٢ – العمليات الحسابية.

٣ – التعابير العلائقية.

٤ – المعامل NOT.

٥ – المعامل AND.

٦ – المعامل OR.

في حالة التكافؤ في الأولوية، فإنّه يتمّ التنفيذ من اليسار إلى اليمين. والأمثلة الآتية توضح طريقة حل التعابير المنطقية التي تحتوي على أكثر من معامل، مع توضيح الأولوية في حل هذه التعابير.

مثال (١)

```
7+1 >= -8  &&  2 <= 1  ||  5 == 10-5
8 >= -8  &&  2 <= 1  ||  5 == 5
True  &&  False  ||  True
False  ||  True
True
```

مثال (٢)

```
7-3 >= 10 ||  2*2 <= 1  &&  5 == 5
4 >= 10 ||  4 <= 1  &&  5 == 5
False  ||  False  &&  True
False  ||  False
False
```

مثال (٣)

يوضح هذا المثال ناتج العبارة المنطقية ((False || True) && False).

```
(False  ||  True) && False
True  &&  False
False
```

مثال (٤)

يوضح هذا المثال ناتج العبارة المنطقية ((False || True) && !False).

```
(False  ||  True) && !False
True  && !False
True  &&  True
True
```

**ثالثا : البيانات المنطقية**

**مهم : تعتبر التعابير العلائقية والتعابير المنطقية جزءاً أساسياً في كثير من البرامج و من الممكن كتابة هذه التعابير مباشرة داخل البرنامج أو تستخدم الكلمة المحجوزة ( bool) للدلالة على نوع البيانات المنطقية وقيمتها إما ( true) أو ( false)**

**** يوضح الشكل الآتي كيفية تمثيل العبارات المنطقية**

```
#include <iostream>
using namespace std;
int main()
{
   bool a = true, b = false, c = true;
   // First: write full Relational Expression
   if( 2 != 5 || 4 + 1 > 4 * 4  &&  5 > 12/3 );
   // Second: write Logic Expression
   if( true || false && true);
   // Third: use bool variables
   if( a || b  && c );
   return 0;
}
```

يوضّح هذا البرنامج طريقة تمثيل العبارات المنطقية:

الأولى: عن طريق كتابة التعابير العلائقية ومعها المعاملات المنطقية.

الثانية: كتابة التعابير المنطقية واستخدام عبارتَي الصواب (true) والخطأ (false).

الثالثة: استخدام المتغيرات المنطقية للتعبير عن عبارتي الصواب (true) والخطأ (false).

وستظهر شاشة المخرجات فارغة لعدم استخدام جملة الطباعة (cout) في هذا البرنامج، وخصوصا بأن البرنامج يقوم بالتحقق من صحة العبارات المنطقية فقط، دون أن يؤدي أيّ وظيفة.

# الوحدة الثانية: البرمجة بلغة (C++)

## الفصل السابع : جملة الاختيار الشرطية

### أولا : جملة الاختيار الشرطية if statement

** تستخدم جملة ( if) : في حالة وجود جملة أو أكثر يرغب المبرمج تنفيذها إذا كانت قيمة التعبير العلائقي أو المنطقي صوابا

❖ الصيغة العامة لجملة ( if):

```
if (condition)
statement;
```

حيث إن:

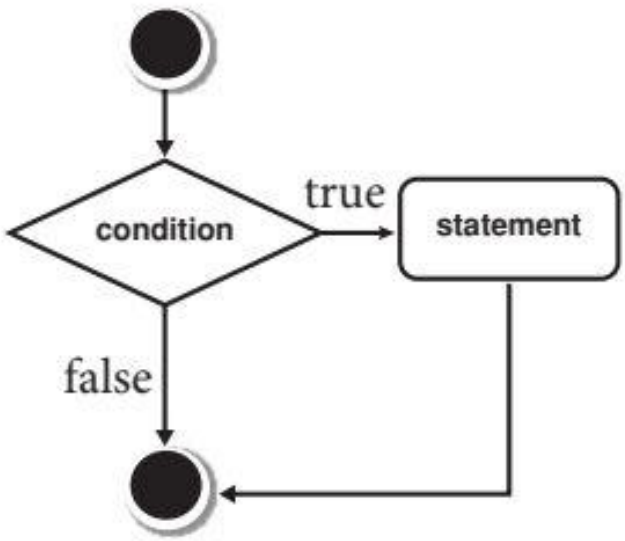
if: كلمة محجوزة من كلمات لغة C++ تعني إذا.

condition: الشرط (تعبير علائقي أو منطقي) قيمته صواب أو خطأ. فلابد من وضعه بين قوسين.

statement;: جملة من جمل C++ تنفذ إذا كانت قيمة الشرط صائبة.

مهمة : عند تنفيذ جملة if فإنه يتم إيجاد القيمة المنطقية للشرط( condition) فإذا كانت قيمته صائبة ( True) فسيتم تنفيذ جملة ( statement1)

** يبين الشكل الآتي مخطط سير العمليات لجملة الاختيار الشرطية if



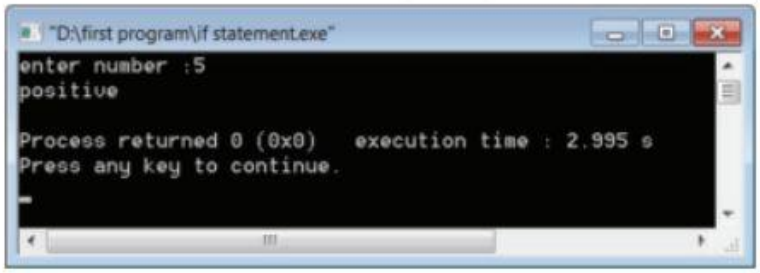
**** مثال : يوضح عملية إدخال قيمة ويحدد إذا كانت القيمة المدخلة موجبة :**

يوضح البرنامج الآتي عملية إدخال قيمة للمتغير num ، فإذا كانت قيمته موجبة يطبع كلمة "positive" :

```
#include <iostream>
using namespace std;
int main()
{
 int num;
  cout<<"enter number :";
  cin>>num;
  if (num>=0)
  cout<<"positive"<<endl;
    return 0;
}
```

١٢٩

ناتج التنفيذ عند إدخال قيمة موجبة هو:



**** يوضح الشكل : برنامجاً يقوم بإدخال عدد، ويطبعه إذا كانت قيمته أكبر من 20 وأقل من 100:**

```
#include <iostream>
using namespace std;
int main()
{
int num;
cin>>num;
if (num>20 && num<100) cout<<num<<endl;
return 0;

}
```

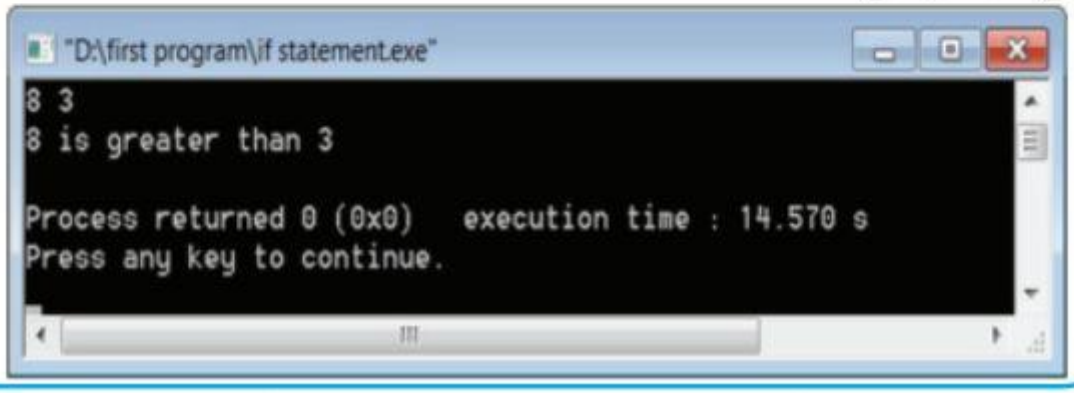
**تنبيه : في حالة وجود أكثر من جملة يرغب المبرمج في تنفيذها في حالة صواب قيمة الشرط تحصر الجمل بين الرمزين {} نوضحها في المثال الآتي :**

```
if (num>=0)
{
cout<<"positive"<<endl;
num=++num;
}
```

**** يوضح الشكل : برنامجاً يقوم بإدخال عددين x,y فإذا كان x أكبر من y ، يطبع x ثم يطبع عبارة "is greater than" ، ثم يطبع y ، ويطرح من قيمة x واحد**

```
#include <iostream>
using namespace std;
int main()
{
 int x,y;
cin>>x>>y;
if (x>y)
{
cout<<x<<" is greater than "<<y<<endl;
x=--x;
}
return 0;
}
```

ناتج تنفيذ البرنامج:




**ثانيا: جملة الاختيار الشرطية المركبة if …else statement**

**** لماذا تستخدم if …else statement : عندما يكون هناك جمل يجب أن تنفذ في حالة صواب قيمة الشرط وجمل أخرئ تنفذ عندما تكون قيمة الشرط خاطئة**

** الصيغة العامة لجملة الاختيار الشرطية if …else:

```
if (condition)
statement1;
else
statement2;
```

حيث إن:

if: كلمة محجوزة من كلمات لغة C ++ تعني إذا.

condition: الشرط (تعبير علائقي أو منطقي) قيمته صواب أو خطأ.

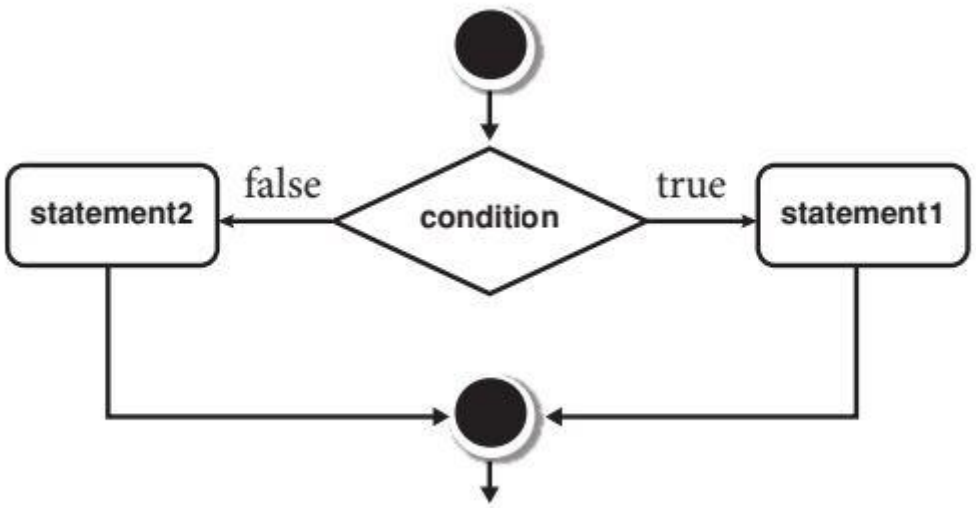
statement1;: جملة من جمل C++ تنفذ إذا كانت قيمة التعبير صائبة.

else: كلمة محجوزة من كلمات لغة C++ تعني إذا لم يتحقق الشرط السابق.

statement2;: جملة من جمل C++ تنفذ إذا كانت قيمة التعبير المنطقي خاطئة.

تنبيه : عند تنفيذ جملة if فإنه يتم إيجاد قيمة الشرط ( condition) فإذا كانت قيمة صائبة فسيتم تنفيذ جملة (Statement 1) وأما إذا كانت قيمته خاطئة فسيتم تنفيذ جملة (statement 2) نوضحها في المثال الآتي :

** يوضح الشكل الآتي مخطط سير العمليات الجملة الاختيار الشرطية if…else

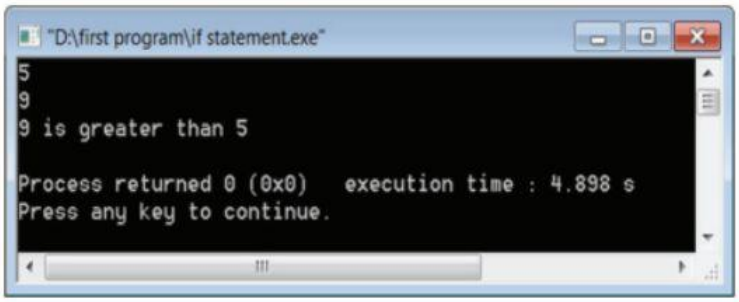


** يوضح الشكل الآتي:برنامجا يقوم في بإدخال عددين غير متساوين x ,y ، فإذا كان x أكبر من y ، يطبع x ثم يطبع عبارة "is greater than" ، ثم يطبع y ، وإذا كان y أكبر من x يطبع y ثم يطبع عبارة " is greater than" ثم يطبع x

```
#include <iostream>
using namespace std ;
int main()
{
int x,y;
cin>>x>>y;
if (x>y)
cout<<x<<" is greater than "<< y<<endl;
else
cout<<y<<" is greater than "<<x <<endl;
 return 0;
}
```

١٣٥

ناتج تنفيذ البرنامج:



# الوحدة الثانية: البرمجة بلغة (C++)

## الفصل الثامن : جملة التكرار for statment

*ملاحظة *

**جملة التكرار for statement : هي إحدى أشهر الجمل التكرار وتستخدم لتكرار تنفيذ جمل لغة C++ بعد محدد من المرات**

### أولا : الصيغة العامة لجملة التكرار for statement

```
for  (Variable_name= initial_value ; condition; step)
 {
Statement 1;
 …
Statement n;
 }
```

حيث إن:

for : كلمة محجوزة من كلمات لغة C++ تعلن عن بدء جملة التكرار.

Variable_name: هو اسم متغير، يجب الالتزام بشروط اختيار اسم المتغير.

initial_value: القيمة الابتدائية للعَدّاد، قد تكون قيمة عددية او متغير عددي أو تعبير حسابي.

ويمكن تعريف المتغير في هذه الخطوة مثل: int counter=1.

condition: شرط (تعبير علائقي أو منطقي) الدخول إلى جمل التكرار.

step : الزيادة الدورية لقيمة العدّاد (الإجراء الذي يحدث بعد تنفيذ الجمل المراد تكرار تنفيذها)، وقد تكون موجبةً أو سالبة، عدداً صحيحا أو عدداً عشرياً، تعبيراً حسابياً أو متغيراً عددياً. وعادة يستخدم معامل الزيادة (++) أو معامل النقصان (--).

{: رمز بداية جملة التكرار، وتستخدم في حال وجود أكثر من جملة يُراد تكرارُ تنفيذها.

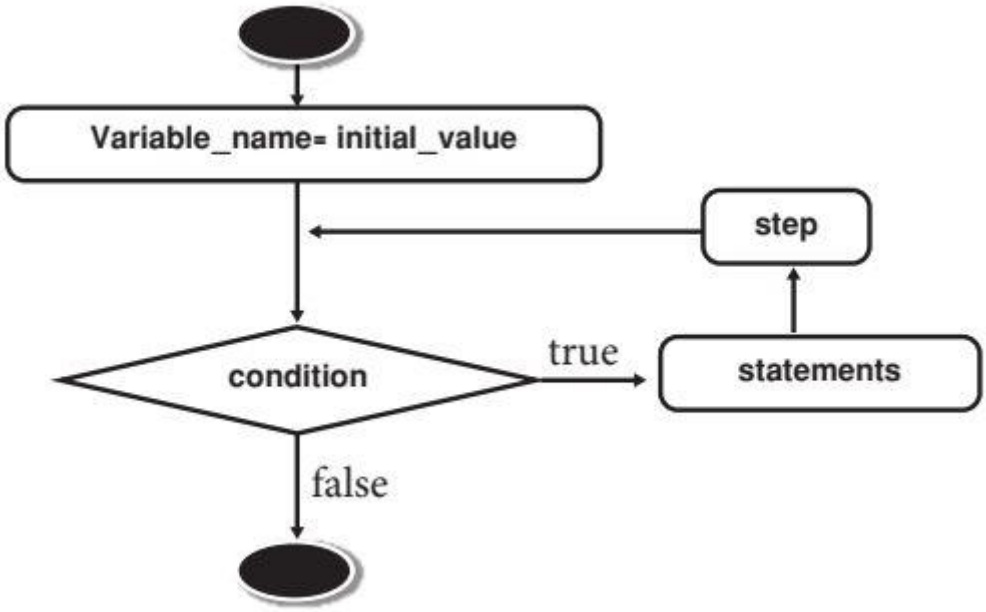
statement 1; … statement n: جملُ C++ التي نرغب بتكرار تنفيذها.

}: رمز نهاية جملة التكرار.

**تنبيه**

1. عند تنفيذ جملة التكرار for يأخذ العداد القيمة الابتدائية ثم يتحقق من الشرط ( condition) فإذا كان كانت قيمة الشرط صائبة يتم تنفيذ الجمل المراد تكرار تنفيذها ثم تعدل قيمة العداد حسب قيمة الزيادة أو النقصان المحددة ويتحقق من قيمة الشرط ( condition)
2. فإذا كانت القيمة صائبة : ينفذ الجمل المراد التكرار تنفيذها ويعدل قيمة العداد بمقدار الزيادة أو النقصان المحددة إلي ان تصبح قيمة الشرط خاطئة

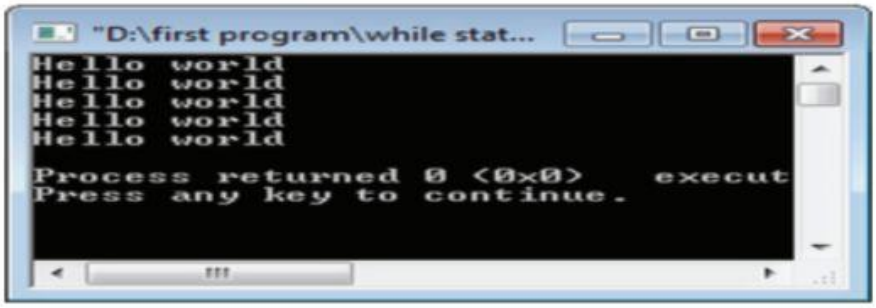
**يبين الشكل الآتي مخطط سير العمليات لجملة التكرار for



**** يوضح الشكل الآتي : برنامجاً يطبع "HeIIo world" باستخدام جملة التكرار (for) خمس مرات**

```
#include <iostream>
using namespace std;
int main()
{
  int counter;
 for (counter =1 ; counter <=5 ; counter ++)
   cout << "Hello world"<< endl;
   return 0;
}
```

ناتج تنفيذ البرنامج:

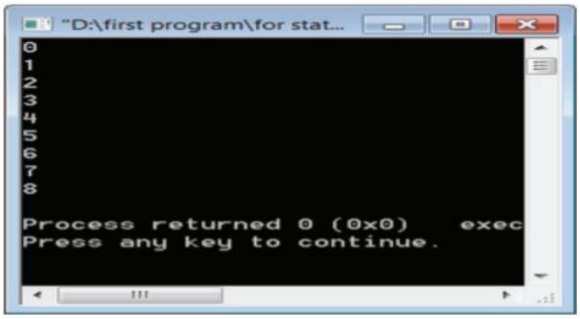



لاحظ أنه عندما تكون قيمة العداد counter أقلّ أو يساوي ٥ يطبع "Hello world" ويتوقف التكرار عندما تصبح قيمة العداد ٦. إذن قيمة العداد counter عند توقف التكرار هي ٦ (وهي أول قيمة تجعل قيمة الشرط خاطئة).

**** يوضح الشكل الآتي: برنامجاً يطبع الأعداد من 0 إلى 8 باستخدام جملة التكرار (for)**

```
#include <iostream>
using namespace std;
int main()
{
 for (int i=0 ; i<=8 ; i++)
   cout << i<< endl;
   return 0;
}
```

ناتج تنفيذ البرنامج:



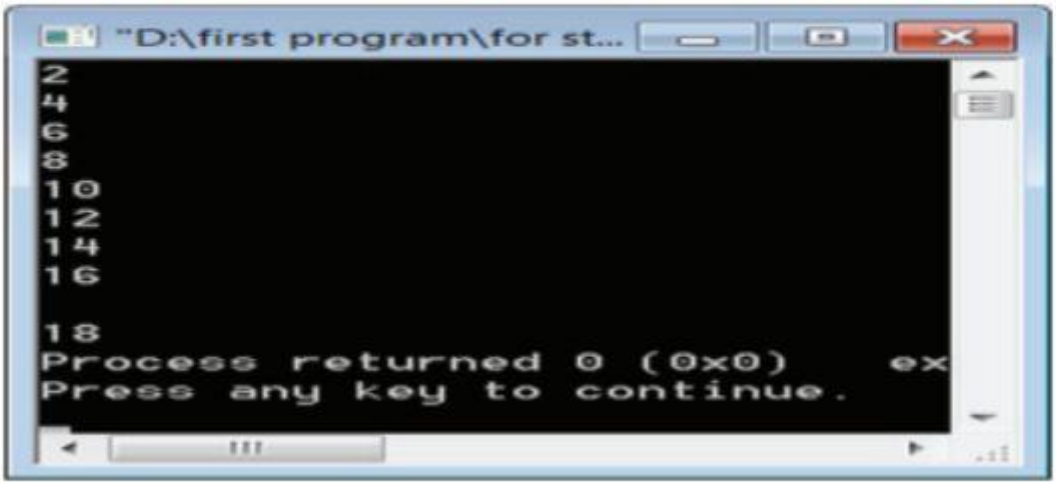

لاحظ عندما تكون قيمة العداد (i) أقل أو يساوي (8) يطبع قيمته، ويتوقف التكرار عندما تصبح قيمة العداد (9). إذن قيمة العداد (i) عند توقف التكرار هي (9).

**** اكتب برنامجاً يطبع الأعداد الزوجية من 2 إلي 16، باستخدام جملة التكرار for**

```
#include <iostream>
using namespace std;
int main()
{
  int j;
  for( j=2 ; j<=16 ; j+=2)
    cout<<j<<endl;
 cout<<endl;
 cout<<j;
   return 0;
}
```

١٤٦

ناتج التنفيذ:

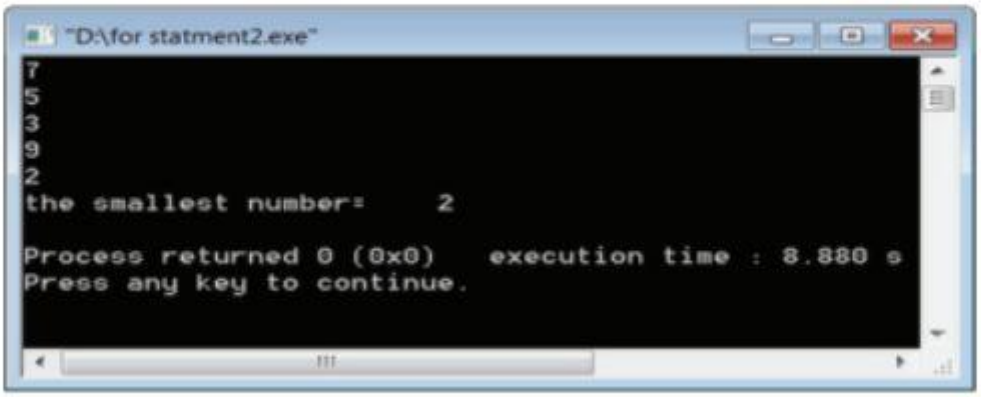



لاحظ أنه تم طباعة قيمة العداد النهائية بعد توقف جملة التكرار وهي (18).

**اكتب برنامجاً يقوم بإدخال 5 أعداد وطباعة العداد الأصغر باستخدام جملة التكرار for

```
#include <iostream>
using namespace std;
int main()
{
int i,min,x;
  cin>>x;
  min=x;
  for (i=1 ; i<5 ; ++i)
   {
    cin>>x;
    if (x<min)  min=x;
   }
  cout<<"the smallest number=    "<<min<<endl;
   return 0;
}
```

ناتج التنفيذ:

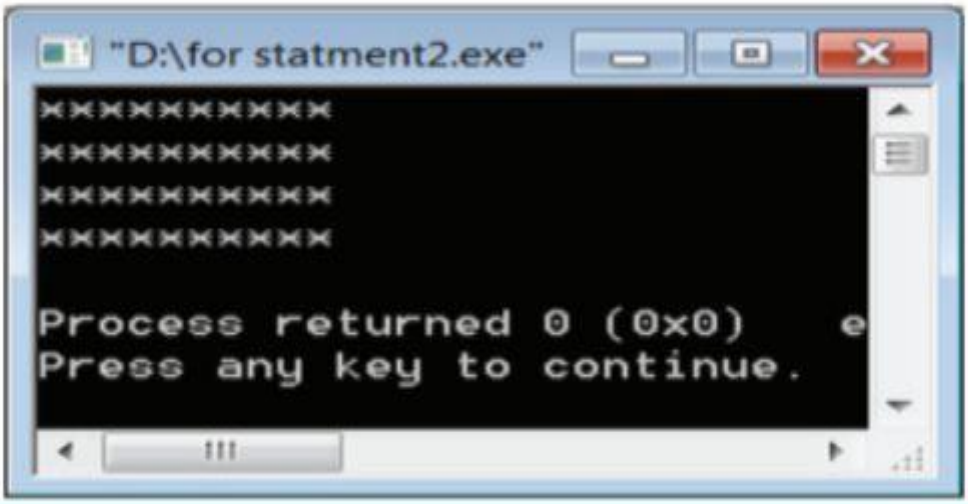



لاحظ أنه تم طباعة أصغر عدد أُدخِل .

** اكتب برنامجاً لطباعة الشكل الآتي باستخدام جملة التكرار for

```
**********
**********
**********
**********
```

```
#include <iostream>
using namespace std;
int main()
{
 int c;
    for (c=1 ; c<=4 ; ++c)
     cout<<"**********"<<endl;
   return 0;
}
```



## ثانيا : المجاميع الإجمالية

### خطوات التي يتبعها الحاسوب لتحقيق المجاميع الإجمالية :

**أولا**
تعريف المتغير الذي سيتم تخزين ناتج عملية الجمع فيه، وإعطائه قيمة أولية هي صفر، حتى لا تتأثر عملية الجمع بأي قيمة قد تكون موجودة داخل المتغير، مثلا int sum = 0 .

↓

**ثانيًا**
البدء بجملة التكرار وإعطاء العداد القيمة الابتدائية، وتحديد الشرط وقيمة الزيادة الدورية على العداد.

↓

**ثالثًا**
أدخل قيمة واحدة للمتغير، مثلا cin>> x ;

↓

**رابعًا**
اجعل القيمة الجديدة للمجموع تساوي القيمة القديمة له مضافاً إليها القيمة المُدخَلة للمتغير، مثلًا
Sum=Sum + x;

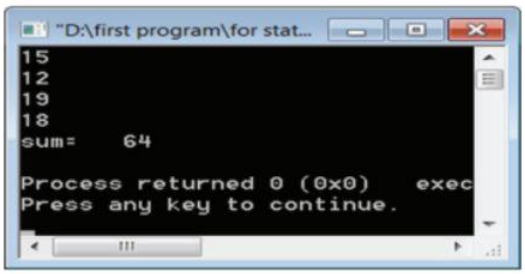
↓

**خامسًا**
كرر ابتداءً من الخطوة الثالثة حتى تتوقف جملة التكرار

### ** اكتب برنامجاً يقوم بإدخال أربع علامات وإيجاد مجموعهم وطباعته :

```
#include <iostream>
using namespace std;
int main()
{
int i,x,sum=0;
   for (i=1 ; i<=4; ++i)
   {
      cin>>x;
      sum=sum+x;
   }
 cout << "sum=\t"<<sum<< endl;
   return 0;
}
```

ناتج التنفيذ:



** اكتب : برنامجاً يقوم بإيجاد المتوسط الحسابي لعلامات طلبة صف مكون 30 طالبا وطباعته

```
#include <iostream>
using namespace std;
int main()
{
   int c,mark,sum=0;
   float average;
   for (c=1 ; c<=30 ; ++c)
   {
      cin>>mark;
      sum=sum+mark;
   }
 average=sum/30;
 cout << "The average=   "<<average<< endl;
  return0;
}
```

** اكتب : برنامجاً يقوم بإيجاد مجموع نواتج القسمة الأعداد من 1 الي 15 على العدد 2 وطباعته

$$\sum_{j=1}^{15} \frac{J}{2}$$

```
#include <iostream>
using namespace std;
int main()
{
   int J,sum=0;
   for (J=1 ; J<=15 ; ++J)
      sum=sum+J/2;
  cout<<sum<<endl;
    return 0;
}
```

**** اكتب : برنامجاً يقوم بإيجاد قيمة المتسلسلة الآتية وطباعتها**

$3 + 6 + 9 + 12 + \ldots + m$

```
#include <iostream>
using namespace std;
int main()
{   int m,c,sum=0;
  cin>>m;
   for (c=3 ; c<=m ; c+=3)
      sum=sum+c;
  cout<<sum<<endl;
    return 0;
}
```

**** اكتب : برنامجاً يقوم بإيجاد قيمة متسلسلة الأتية وطباعتها**

$$\frac{1}{2}+\frac{1}{4}+\frac{1}{6}+\ldots+\frac{1}{n}$$

```
#include  <iostream>
using namespace std;
int main()
{
 int n;
 float i,sum=o;
 cin>>n;
 for (i=2 ; i<=n ; i+=2)
   sum=sum+1/i;
 cout<<sum<<endl;
    return 0;
}
```

**وفقكم الله لما يحب ويرضى**

قروب و صفحة المعلمة: نعمة الأخرس

https://www.facebook.com/groups/4302651726462421/?ref=share

https://web.facebook.com/nemehmohieb

صفحة تلاخيص منهاج أردني [كامل دروس المنهاج الأردني تلاخيص وشروحات]

https://web.facebook.com/talakheesjo

ملفاتنا على التيليجرام
https://t.me/talakheesjo