

C++

- `Int main() {}` هي الدالة الاساسية والتي يكتب فيها كل الاوامر
ملحوظة : الاوامر تكتب داخل

{ }

- يجب وضع ; في نهاية كل امر لتعبير عن انتهاء الكود
 - لطباعة كلمة او حرف او رقم نستخدم امر `cout<<` وهي اختصار ل `cout output`
- المكتبة : هي مجموعة من الاوامر الجاهزة للاستخدام وتطلب استدعاء لاستخدامها ومنها مكتبة `iostream` التي يوجد بها امر `cout`

كيفية استدعاء مكتبة `iostream` واستخدام امر `cout`:

1. يجب وضع امر الاستدعاء `<include>`
 2. تكتب اسم المكتبة داخل `<>` مثل `<iostream>`
- ملحوظة : يجب وضع امر الاستدعاء في اول سطر
3. يجب كتابة امر ; `using namespace std` لاستخدام امر `cout`
 4. عند وضع كلمات او جمل ف امر `cout` يلزم وضع " " ولاكن عند استخدام الارقام يمكن وضعها بدون " "
- يمكن طباعة اكثر من مخرج باستخدام `cout` واحدة مثل :

```
Cout <<"hello"<<"world";
```

بدل من

```
Cout<<"hello";
```

```
Cout<<"world"
```

- يمكن طباعة سطر جديد باستخدام اكثر من طريقة منها :
1. طباعة امر `endl` وهو اختصار ل `end line` كتالي :

```
Cout<<endl ;
```

او

```
Cout<<"hello"<<endl ;
```

2. استخدام \n ويلزم وضعها في " "

- لتترك اربع مسافات نستخدم الامر \t وهي تعادل ذر tap
- ال comment او التعليق هو صيغة يتم استخدامها لوضع كلام الكمبيوتر غير قادر على قرائتها وتستخدم لوضع ملاحظات للمبرمج الذي يكتب الكود او لمبرمج اخر يطلع عليه
- لانشاء تعليق نكتب // في بداية السطر ولاكن لسطر واحد فقط وعند استخدام اكثر من سطر نستخدم /* */ ونضع التعليق في المنتصف

ملاحظة: التعليق يمكن استخدامه في الغاء كود لفصل الاكواد وتجربة صحتهم واحدا تلو الاخر

- Return 0 ; هو كود يوضع في اخر دالة main ويعبر عن انتهاء الكود بشكل صحيح وبدون اخطاء واى امر يكتب بعده لاينفذ لانه يعبر عن انتهاء البرنامج

ملحوظة: يمكن تغيير الرقم الذى بعد return ولاكن الاكثر استخداما هو 0

مثال :

```
#include<iostream>
```

```
Using namespace std ;
```

```
Int main (){
```

```
Cout<<"hello" ;
```

```
Cout<<"world" ;
```

```
Return 0 ;
```

```
Cout<<"bye"
```

```
}
```

سيقوم الكود بطباعة hello وطباعة world ولن يطبع bye لانها بعد امر return

المتغيرات

هو مكان في الذاكرة يتم تخزين فيه قيمة يمكن ان تكون هذه القيمة (عدد,حرف,كلمة,جملة, false,true)

كيفية اضافة المتغير :

1. كتابة نوعه
2. اسم المتغير
3. = قيمته (يمكن ان تكتبها او لا)
4. ;

مثال :

```
int x=22 ;
```

انواع المتغيرات :

| نوع المتغير | تعريفه | مثال |
|-------------|-------------------------------|---------------------------|
| Int | هو متغير لحفظ الارقام الصحيحة | 2 , 3 , 5 , 24 , 140 , 99 |
| String | هو متغير لحفظ النصوص | "hello","ahmed" |
| Double | هو متغير لحفظ الارقام الكسرية | 8.0 , 5.1 , 2.3 |
| Float | هو متغير لحفظ الارقام الكسرية | 8.0 , 5.1 , 2.3 |
| Char | هو متغير لحفظ الحروف | 'A' , 'a' |
| Bool | هو متغير لحفظ true او false | True false |

ملاحظات :

- لايلزم استخدام " " عند وضع رقم
- يلزم وضع " " عند استخدام الكلمات او الجمل
- ال double يتمكن من استيعاب 14 رقم بعد العلامة العشرية
- ال float يتمكن من استيعاب 7 ارقام بعد العلامة العشرية
- عند استخدام حرف نضعه بين علامة ''
- عند طباعة true او false يتم طباعتهم 1 و 0 حيث true تعنى 1 و false تعنى 0
- عند كتابة اكثر من متغير يجب مراعاة ان تكون الاسماء متخلفة
- عند تغيير قيمة متغير تمت اضافة قيمته بالفعل نقوم بكتابة اسم المتغير مباشرة بدون كتابة نوعه ثم نكتب = ثم القيمة الجديدة
- يمكن اضافة اكثر من متغير في فس الوقت كالتالى :

```
Int x=1 , y=2 , z=3 ;
```

- لا يجب ان يكون اسم المتغير مبدوء برقم كتالى : 7o7o , 7osam ولاكن يمكن كتابة الرقم من ثانى خانة او اكثر مثل : mo7md , a7md

ادخال قيمة

- يتم استخدام امر >>cin ويتم اتباعه باسم المتغير المراد وضع القيمة فيه
 - يلزم قبل وضع اسم المتغير تعريف المتغير
- يمكن عمل اكواد بسيطة عند الوصول الى هذا الحد مثل :
1. كود الترحيب للمستخدم بواسطة اسمه
- كتالى :

```
#include<iostream>
using namespace std;

int main (){
cout<<"what is your name"<<endl; // رسالة لاستقبال الاسم
string user ; // انشاء متغير لتخزين فيه اسم المستخدم
cin>>user ; // استقبال الاسم من المستخدم
cout<<"welcome "<<user<<" in the calcluter"; // طباعة رسالة الترحيب مستخدما اسم المستخدم
}
```

2. آلة حاسبة تقوم بعملية واحدة فقط كتالى :

```
3. #include<iostream>
4. using namespace std;
5.
6. int main (){
7. cout<<"enter the frist number"; // امر طباعة طلب ادخال الرقم الاول
8. double num1 ; // انشاء المتغير الذى سيحفظ فيه الرقم الاول
9. cin >> num1 ; // امر استقبال الرقم الاول
10.cout<<"enter the next number" // امر طباعة ادخال الرقم الثانى
11.double num2 ; // انشاء المتغير الذى سيحفظ فيه الرقم الثانى
12.cin >> num2 ; // استقبال ارقم الثانى
13.cout <<num1 + num2 ; // امر الجمع (يمكن تغيير العملية بتغيير ال + الى * , / , % )
14.}
```

- في هذا المثال قمنا بعمل عملية الجمع واذا اردنا تغيير العملية نقوم فقط بتغيير علامة + الى
1. (-) فى حالة الطرح
 2. (*) فى حالة الضرب

3. (/) فى حالة القسمة

- تم استخدام نوع المتغير double وليس int لمراعاة اذا كانت العملية قسمة وكانت توجد هناك عمليات حسابية ناتجة رقم عشري

ملاحظة : يمكن عمل اى عملية حسابية داخل ال cout ولاكن عند استخدام الارقام

خارج القسمة: (modulus)

هو واحد من العمليات الحسابية مثل الضرب والجمع والطرح والقسمة وله حلتان

1. Small %big

وفى هذه الحلة يكون خارج القسمة هو الرقم الصغير
مثال :

- $4\%5=4$
- $6\%11=6$
- $1\%2=1$

2. Big % small or number % the same number

وفى هذه الحالة نقوم بالتفكير فى الرقم الذى سنزيله من الرقم الاول لتصبح القسمة بدون اظهر كسر
مثال :

- $5\%4=1$
- $15\%7=1$
- $2\%2=0$
- $9\%3=0$
- $12\%5=2$

- فى المثال الاول والثانى اذا قمنا بعمل قسمة سينتج رقم عشري (2.143 , 1.25) لذلك عند التفكير فى ما ارقم الذى يجب ان ازيله لتصبح الارقام صحيحة سيكون 1 فاذا قمت بازالة 1 من 5 فى المثال الاول سيكون ناتج القسمة 1 عدد صحيح وفى المثال الثانى اذا قمت بازالة 1 من 15 ستصبح 14 وعند ذلك ستقسم على 7 ويكون الناتج 2 عدد صحيح
- فى المثال الثالث والرابع نجد انه لا يوجد رقم نستطيع اذالته ليصبح الناتج صحيح لانه رقم صحيح بالفعل حيث فى المثال الثالث عند القسمة نجد الناتج 1 رقم صحيح وعند المثال الرابع نجد الناتج 3 رقم صحيح لذلك لانحتاج لازالة اى رقم وبتالى يصبح خارج القسمة 0

جمع او طرح متغير يحمل قيمة عددية :

اذا كان هناك متغير قيمته 5 ($x=5$) اذا ارت ان اقوم بزيادة او انقاص قيمة هذا المتغير يمكن عمل ذلك باكثر من طريقة .

1. باستخدام رقم كتالى : $x+3=8$

2. باستخدام متغير اخر يحمل قيمة اخري كتالى :

$Z=8$

$X+Z=13$

3. باستخدام خاصية ++ او --

ملاحظة : هذه الخاصية تقوم بجمع او طرح واحد فقط كتالى :

$x++=6$

$x--=4$

يرجى استخدام هذه الخاصية فى سطر منفصل ولا يفضل اشراكها فى اكواد اخري

كتالى :

`Int x =1;`

`X++`

`Cout<<x`

وليس كتالى :

`Int x =1`

`Cout<<x++`

لانه سيقوم بطباعة قيمة x اولا قبل الجمع فايفضل كتابتها كتالى :

`Int x =1`

`Cout << ++x`

تسمى : $x++$ ب postfix و $++x$ ب prefix

العمليات الحسابية تتم بالقوانين الرياضية حيث ما بداخل الاقواس اولا وان الضرب والقسمة وخارج القسمة يسبقو الجمع والطرح

المقارنة :

نستخدم فى المقارنة (`&&` , `||` , `<=` , `>=` , `<` , `>` , `=` , `==`)

ملاحظة :

- الفرق بين = و == هو ان = معناه قيمة تساوي قيمه بينما == تدل ل السؤال عن هل تتساوى القيمتان وفي حالة نعم يكون الناتج 1 ليدل على true والعكس يكون الناتج 0 ويدل على false
- علامة && معناها " و " و علامة || معناها " او "
- يتم استخدام && و || فى التفكير المنطقي حيث ان && يكفى لتحقق الشروط كلها لتعطى قيمة 1 بينما || يكفى شرط واحد لتعطى قيمة 1
- نستخدم ! للنفي حيث != تعنى لا تساوى

if :

كيفية الاستخدام :

- تكتب هكذا if () { }
 - داخل () تكتب الشروط
 - واذا تحققت الشروط يتم تنفيذ الموجود داخل { }
- مثال:

```
Int a =30 ;
If (a==30)
{
Cout<<"a s not equal 30"

}
```

- وفي حالة عدم تحقق الشرط لا يتم عمل الموجود داخل { }
 - عند اضافة وجود احتمال اخر نستخدم if () { } else ولاكن يجب ان يكون اول احتمال ب if وباقي الاحتمالات ب else if
 - فى حالة وضع عمل ينفذ اذا لم تتحقق جميع الاحتمالات نستخدم else { } وهى تكون بدون () ويلزم اضافتها فى النهاية بعد كتابة كافة الاحتمالات
 - يمكن عمل جملة if بطريقة اخرة كتالى :
(a==30)?cout<<"a is 30":cout a is not 30 ;
 - قمت بعمل شرط داخل () ثم نضيف ؟ ثم نكتب الامر المراد تنفيذه فى حالة تحققه ثم اكتب : والامر المراد تنفيذه فى حالة تحقق الشرط
 - يمكن كتابة if داخل if
 - يمكن البدء بعمل مشاريع بالوصول الى هذا الحد مثل :
1. الة حاسبة

```

2. cout<<"hello to the calculter"<<endl;
3. cout<<"chouse the numper of opration"<<endl<<"1  +"<<endl<<"2  -
   "<<endl<<"3  *"<<endl<<"4  /"<<endl;
4. int x ;
5. cin>> x ;
6. int num1 ;
7. int num2 ;
8. if(x==1){
9.     cout<<"enter the frist number";
10.    cin>>num1;
11.    cout<<"enter the secound number";
12.    cin>>num2;
13.    cout <<"the result is : "<<num1+num2 ;
14.    }
15.else if (x==2)
16.{
17.    cout<<"enter the frist number";
18.    cin>>num1;
19.    cout<<"enter the secound number";
20.    cin>>num2;
21.    cout <<"the result is : "<<num1-num2 ;
22.}
23.else if (x==3)
24.{
25.    cout<<"enter the frist number";
26.    cin>>num1;
27.    cout<<"enter the secound number";
28.    cin>>num2;
29.    cout <<"the result is : "<<num1*num2 ;
30.}
31.else if (x==4)
32.{
33.    cout<<"enter the frist number";
34.    cin>>num1;
35.    cout<<"enter the secound number";
36.    cin>>num2;
37.    cout <<"the result is : "<<num1/num2 ;
38.}
39.else {
40.    cout<<"you enter a wrong number please try again" ;
41.}
42.

```

- اداة switch هى اداة مثل if تستخدم لعمر اوامر فى حالات تابعا لمتغير كتالى


```
Switch(x){
```

```
Case 3 :
```

```
    Cout <<"x is 3";
```

```
    Break;
```

```
Case 4:
```

```
    Cout<<"x is 4";
```

```
    Break;
```

```
Default:
```

```
    Cout<<"x is not 3 or 4";
```

```
}
```

- المتغير الذى تكون الاحتمالات تابعة له يكتب داخل ()
- عند كتابة احتمال نكتب case ثم قيمة المتغير ثم : ثم الاوامر المراد فعلها اذا تحقق هذا الاحتمال
- ونهى الاحتمال بكتابة break ; (مهمة جدا)
- لوضع امر فى حالة عدم تحقق الاحتمالات نستخدم default

For

هى اداة لتكرار مثال :

عند الحاجة لطباعة رسالة مرة يمكن استخدام cout ولاكن عند الحاجة لطباعة الرسالة 100000 مرة هنا نستخدم ال loop ومنها for loop

- تكتب كالتالى :

```
For () {}
```

- داخل () يكتب متغير وقيمه وشرط استمرار ال loop والعمليات التى ستنتم عليه
- داخل {} تكتب العمليات التى ستنتم
- ال loop يعيد ويكرر نفسه الى ان يتحقق الشرط الموجود داخل () لذلك من الخطير عدم وضع نهاية لل loop لانه سيجعل الكمبيوتر لايتوقف مما سيؤدى الى اتلاف الجهاز
- مثال عند طباعة كلمة hello 50 مرة نكتب

```
• for (int i = 0; i <11; i++)
```

```
• {  
•   cout<<"hello";  
• }  
•
```

- سيقوم الكود بعمل متغير اسمه `i` ويضع قيمته ب `0` ويحقق من شرط انه اقل من `11` ثم يقوم بزيادة ال `i` بمقدار `1` وينفذ الامر وهو طباعة `hello`