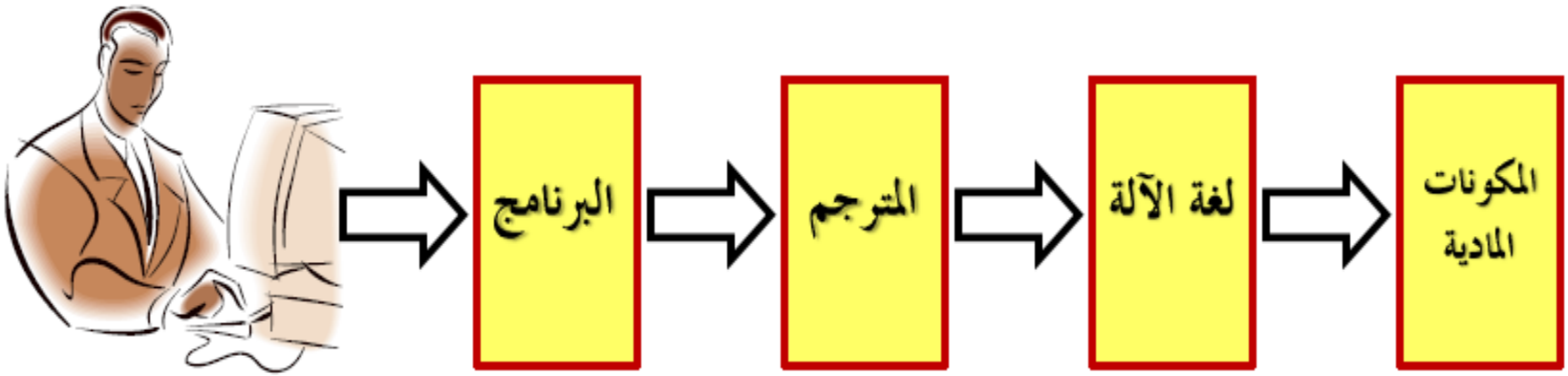
جامعة حماة
كلية العلوم في مصياف / السنة الأولى

المادة: البرمجة و الخوارزميات

المحاضرة الثالثة: حلقات التكرار في لغة البرمجة C++ Loops

العام الدراسي ٢٠١٨ - ٢٠١٩



جمل الدوران في لغة C++

- جمل التكرار او الدوران التي نستخدمها في كل لغات البرمجة تكاد تتطابق مع مثيلاتها في لغة C++، سواء في المفهوم أو طريقة الاستخدام.
  - for ✓
  - while ✓
  - do… while ✓
  - For each ✓ مع المصفوفات سنناقشها في محاضرات لاحقة

عبارة For .. Loop التكرارية الاعتيادية

- لقد تعلمنا أن الحاسب يقوم بتنفيذ البرنامج خطوة خطوة (بدون تكرار لأي خطوة حتى يصل إلي نهاية البرنامج) وتعلمنا أنه في حال وجود جملة if الشرطية في البرنامج يمكن استثناء بعض الخطوات من التنفيذ إذا لم يتحقق الشرط.
- عبارة for: عبارة تكرارية تستخدم لتكرار سطر برمجي واحد في حالة لم نضع أقواس، أو تكرار مجموعة سطور برمجية في حال حصرها بين قوسين لعدد معين من المرات يحددها المبرمج.
وشكلها العام:

```
for (Initializing; Boolean_Expression; Update)
{
statement
}
```

```
for (int i=0; i<100; i++)
{
if (i%2==0)
Cout<<i<<"\t";
 }
```

- **Initializing**: القيمة البدائية المعطاة للمتغير التي منها سيبدأ العد. مثلا (i = 0)
- **Boolean_Expression:** شرط التوقف الذي تصبح عنده قيمة المتغير غير محققة لشرط التوقف وسوف يخرج من عبارة for.
- **Update:** مقدار الزيادة أو النقصان في قيمة المتغير في كل دورة (i-- , i++)
- **Statement:** العبارات البرمجية التي سيتم تنفيذها عدة مرات (بحسب العداد في for).

جملة الدوران for

١. تستخدم جملة for لتكرار مجموعة من العمليات ابتدأ من قيمة ابتدائية إلى قيمة نهائية محددة مسبقاً حسب الشرط المحدد بمقدار زيادة أو نقصان للعداد التي يجب أن تؤدي إلى نهاية محددة وإلا لا ينتهي التكرار أي يتكرر الحدث إلى ما لانهاية مثل :

for (i=1 ; i<=100 ; i++) ---> loop from 1 to 100

for (i=1 ; i<=100 ; i--) ---> loop from 1 to infinity

١. تحديد القيمة الابتدائية للعداد Initialization: يتم في هذا الجزء تحديد القيمة التي يبدأ منها التكرار عن طريق إحلال قيمة عددية للعداد ،العداد هو متغير يجب أن يكون معدود ordinal بحيث يكون من نوع عدد صحيح int أو رمز char .

٢. الشرط condition : وهو تعبير منطقي يجب أن يحتوي على عملية مقارنة للعداد مع قيمة تمثل نهاية التكرار .

٣. مقدار التغير increment : وهو يمثل مقدار زيادة أو نقصان العداد بمقدار معين تبعاً للمشكلة المراد حلها .

٤. تستمر عبارة for بالتكرار مادام جواب الشرط متحقق true وينتهي التكرار عندما يكون الشرط false .

أمثلة على عبارة For .. Loop التكرارية الاعتيادية

١- برنامج يقوم بطباعة الأعداد من (٠ إلى ٩).

```
#include<iostream>
using namespace std;
void main()
{
for (int i=0; i<10; i++)
cout<<i<<"\t";
}
```

النتيجة التي ستظهر على شاشة الحاسب:

٠ ١ ٢ ٣ ٤ ٥ ٦ ٧ ٨ ٩

٢- برنامج يقوم بطباعة الأعداد من (٤ إلى ٠).

```
#include<iostream>
using namespace std;
void main()
{
for (int i=4; i>=0; i--)
cout<<i<<"\n";
}
```

النتيجة التي ستظهر على شاشة الحاسب:

٤
٣
٢
١
٠

أمثلة على عبارة For .. Loop التكرارية الاعتيادية

١- برنامج لطباعة ناتج جمع الأعداد الزوجية الواقعة في المجال {100 – 0 }.

```
#include<iostream>
using namespace std;
void main()
{
int sum=0;
for (int n=0; n<=100; n++)  {
if (n%2==0)
sum += n ; }
cout<<"sum="<<sum;
}
```

النتيجة التي ستظهر على شاشة الحاسب:

Sum=2525

٢- برنامج لطباعة ناتج جمع الأعداد الفردية الواقعة في المجال {100 – 1 }.

```
#include<iostream>
using namespace std;
void main()
{
int n, sum=0;
for (n=1; n<=100; n=n+2)
sum += n ;
cout<<"sum="<<sum;
}
```

النتيجة التي ستظهر على شاشة الحاسب:

Sum=2525

أمثلة على عبارة For .. Loop التكرارية الاعتيادية

- برنامج لطباعة مجموع ومعدل درجات طالب في ١٠ مقررات دراسية.

```
#include<iostream>
using namespace std;
void main()
{
int mark, sum=0;
float avg=0.0;
for (int i=0; i<10; i++)
  {
    cin>>mark;
    sum += mark ;
   }
avg= sum/10;
cout<<"sum="<<sum<<"\n Avg="<<avg;
}
```

النتيجة التي ستظهر على شاشة الحاسب:

Sum=??? Avg= ???

أمثلة على عبارة For .. Loop التكرارية الاعتيادية

- برنامج لطباعة ناتج فحص العدد المدخل (عدد أولي أم لا).

```
#include<iostream>
using namespace std;
void main()
{
int num, bx=0;
cin>>num;
for(int i=2;i<num ;i++)
if( num%i==0)    bx=1;
if (bx==1)
cout<<num<<"The number is no prime" ;
else
cout<<num<<" The number is prime" ;
}
```

النتيجة التي ستظهر على شاشة الحاسب:

? The number is no prime

او

النتيجة التي ستظهر على شاشة الحاسب:

? The number is prime

عبارة For .. Loop التكرارية المتداخلة

- العبارة التكرارية الاعتيادية نحتاجها لتكرار سطر برمجي واحد أو عدة سطور لغرض ما. أما العبارة التكرارية المتداخلة فلا تكرر فقط سطور برمجية إنما تكرر عبارات for **(أو عبارات** تكرارية أخرى **while do—while)**
- العبارة التكرارية الداخلية يمر عليها عند كل زيادة جديدة للعبارة التكرارية الأم لذلك في كل زيادة واحدة للعبارة الأم تزيد العبارة التكرارية الداخلية جميع عداتها الممكنة حسب شرطها. وشكلها العام:

```
for (initializing1; boolean_expression1; update1)
{
for (initializing2; boolean_expression2; update2)
{
statement
}
}
```

- **initializing1,2**: القيمة البدائية المعطاة للمتغير التي منها سيبدأ العد. مثلا (i = 0, J=0)
- **boolean_expression1,2:** شرط التوقف الذي تصبح عنده قيمة المتغير غير محققة لشرط التوقف وسوف يخرج من عبارة for.
- **update1,2:** مقدار الزيادة أو النقصان في قيمة المتغير في كل دورة (i++, J++, i--, J--)
- **statement:** العبارات البرمجية التي سيتم تنفيذها عدة مرات (بحسب العداد في for).

أمثلة على عبارة For .. Loop التكرارية المتداخلة

١- برنامج لطباعة شكل مثلث من النجوم.

```
#include<iostream>
using namespace std;
void main()
{
for (int i=5; i>0; i--)
  {
   for (int j=1; j<=i; j++)
   cout<< " * " ;
   }
 cout<<endl;
}
```

النتيجة التي ستظهر على شاشة الحاسب:

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

٢- برنامج لطباعة شكل مثلث من الأرقام .

```
#include<iostream>
using namespace std;
void main()
{
for (int i=1; i<=5; i++)
  {
   for (int j=1; j<= i; j++)
   cout<< j ;
   }
cout<<endl;
}
```

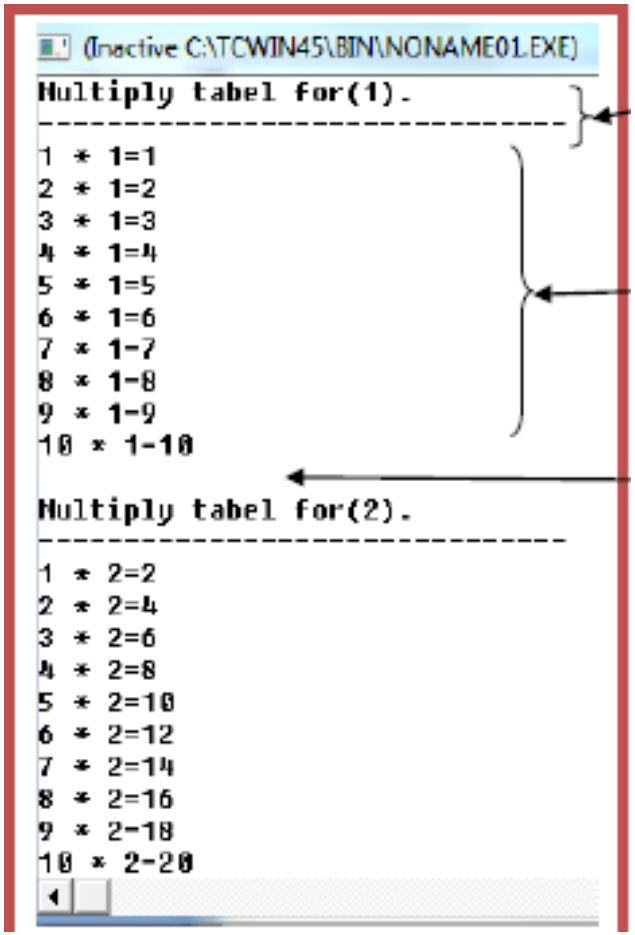
النتيجة التي ستظهر على شاشة الحاسب:

```
1
12
123
1234
12345
```

أمثلة على عبارة For .. Loop التكرارية المتداخلة

١- برنامج لطباعة جداول ضرب الأعداد من (١ – ١٠)

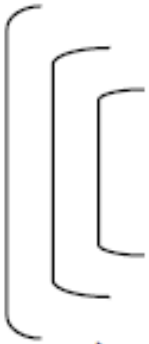
```
#include<iostream>
using namespace std;
void main()
{
for (int i=1; i<=10; i++)
  {
   cout<<"Multiply tabel for("<<i<<“)”;
     cout<<"\n----------------------------\n";
    for (int j=1; j<=10; j++)
    cout<< j<<" * " <<i<<"="<<i*j<<"\n";
   cout<<"\n";
   }
}
```



عبارات التكرار المتداخلة Nested Loop Statement

- كل عبارة من عبارات التكرار الثلاثة التي درسناها تحتوي على جملة أو أكثر من الجمل التابعة لها، هذه الجمل يمكن أن تكون عبارة إحلال أو عبارة شرط ..الخ ، أما إذا كانت واحدة من هذه الجمل هي عبارة تكرار أيضاً فتسمى هذه الحالة بعبارات التكرار المتداخلة . أي أن أي عبارة تكرار تحتـــوي على عبارة تكرار واحدة أو أكثر بداخلها تسمى بالعبارات المتداخلة ، وشرط التداخل هذا هو عدم حصول تقاطع في الحلقات التكرارية المتداخلة كالتالي :

- تداخل ثلاثي غير صحيح (متقاطع)



تداخل ثلاثي صحيح

عبارة break (الخروج/ قطع)

- تستخدم هذه الدالة للتوقف عن العبارات التكرارية عند شرط معين والانتقال إلى ما بعد العبارة التكرارية (لاحظ إلى أين ينتقل السهم). أي انه فقط يخرج من عبارة التكرار الموجود هو فيها وينتقل لينفذ الخطوات البرمجية التي تليه.

```
for (initializing1; boolean_expression1; update1)
{
if (condition)
break;
statement;
}
statement;
```

مثال: طباعة الأعداد من (1-100) والخروج من الحلقة التكرارية عندما يصل العداد لقيمة = 75

```
#include<iostream>
using namespace std;
void main()
{
for (int i=1; i<=100; i++)  {
if (i==75)
break;
cout<<i<<"\t";  }
cout<<"\n is finished print to 75";
}
```

مثال على عبارة break (الخروج)

- إذا أردنا أن لا يتوقف العداد أبدا أو يتوقف بشرط معين (مثلا برنامج التحقق من الرقم السري) حيث سيطلب من المستخدم إدخال الرقم السري حتى يدخل الرقم صحيح، بعدها يخرج من العبارة التكرارية وهذا ابسط مثال على عبارة تكرارية لانهائية.
- **مثال:** برنامج إدخال الرقم السري ويستمر بطلب الرقم من المستخدم إلى أن يدخل الرقم الصحيح

```
#include<iostream>
using namespace std;
void main()
{  int password;
  for (;;)
    {
     cout<<"enter password\n";
     cin>> password;
    if(password==1234)
     {  cout<<"password is correct";
         break;   }
   else
   cout<<"password is error try agin\n";
     }
}
```

عبارة continue(الإهمال والمتابعة)

- تستخدم هذه الدالة لإهمال الخطوات البرمجية التي تليها والانتقال إلى عبارة التكرار ومتابعة العد حسب شرط عبارة التكرار إذا كانت تقبل زيادة جديدة (لاحظ شكل السهم إلى أين ينتقل).أي انه فقط بهمل Statement وينتقل إلى عبارة التكرار لبيداء عدة جديدة اعتيادية

```
for (initializing1; boolean_expression1; update1)
{
if (condition)
continue;
statement;
}
```

مثال: طباعة الأعداد من (1-100) وإهمال طباعة العدد 50.

```
#include<iostream>
using namespace std;
void main()
{
for (int i=1; i<=100; i++)  {
if (i== 50)
continue;
cout<<i<<"\t";  }
cout<<"\nis finished print 100 execpt 50";
}
```

عبارة **go to** (الانتقال إلى)

- تؤدي هذه العبارة إلى انتقال تنفيذ البرنامج إلى سطر معين له عنوان، أي الانتقال إلى السطر الذي يحمل اسم ونضع بعده (:) وهيكلية هذه الدالة كالتالي:

```
main()
{
Statement1;
if (condition)
goto Npoint;
Statement2;
Statement3;
Statement4;
Npoint:
Statement5;
}
```

مثال: طباعة الأعداد من (0-9) بواسطة تكرار تعليمة go to

```
#include<iostream>
using namespace std;
void main()
{
int num=0;
s1:
cout<< num<<"\t";
num=num+1;
if (num!=10)
goto s1;
}
```

جمل الدوران While في لغة C++

الجمل الابتدائية
while (شرط التكرار)
عملية واحدة لتكرارها;

```
#include " stdafx.h"
#include " iostream.h"
main ()
{
 int a;
a=1;
while (a<5)
cout<<a++ <<endl;
return0;
}
```

الجمل الابتدائية
while (شرط التكرار)
{
مجموعة من العمليات لتكرارها
}

انتبه في حال كان هناك أكثر من جملة نريد تنفيذها في كل دورة لابد أن نضعهم جميعا بين قوسين

عبارة while التكرارية

- هي عبارة تكرارية تستخدم لتكرار السطور البرمجية المحصورة بين قوسيها لعدد من المرات تكون مشابه تماما لعبارة التكرار for. ونستطيع المقارنة بين هيكلية هاتان العبارتان التكراريتان وسوف نجد التشابه بينهما:

```
Initializing;      1
while ( Boolean_Expression)    2
{
Statement;         3
Update;            4
}
```

- لإدخال while في حلقة لانهائية نكتب while(1) فقط

- **Initializing**: القيمة البدائية المعطاة للمتغير التي منها سيبدأ العد. مثلا (i = 0)
- **Boolean_Expression:** شرط التوقف الذي تصبح عنده قيمة المتغير غير محققة لشرط التوقف وسوف يخرج من عبارة while.
- **Update:** مقدار الزيادة أو النقصان في قيمة المتغير في كل دورة (i-- , i++)
- **Statement:** العبارات البرمجية التي سيتم تنفيذها عدة مرات (بحسب العداد في for).

أمثلة على عبارة while التكرارية

١- برنامج يقوم بطباعة الأعداد من (٠ إلى ٩).

```
#include<iostream>
using namespace std;
void main()
{
int i=0;
while(i<=9)
 {
   cout<<i<<"\t";
   i++;
 }
}
```

النتيجة التي تظهر على شاشة الحاسب:

٠ ١ ٢ ٣ ٤ ٥ ٦ ٧ ٨ ٩

٢- برنامج يقوم بطباعة الأعداد من (٤ إلى ٠).

```
#include<iostream>
using namespace std;
void main()
{
int i=4;
while(i>=0)
 {
   cout<<i<<"\n";
   i--;
 }
}
```

النتيجة التي تظهر على شاشة الحاسب:

٤
٣
٢
١
٠

أمثلة على عبارة while التكرارية

١- برنامج لطباعة ناتج جمع الأعداد الزوجية الواقعة في المجال {100 – 0 }.

```
#include<iostream>
using namespace std;
void main()
{
int i=0, sum=0;
while(i<=100)
{ sum += i ;
i=i+2; }
cout<<"sum="<<sum;
}
```

النتيجة التي تظهر على شاشة الحاسب:

Sum=2525

٢- برنامج لطباعة ناتج جمع الأعداد الفردية الواقعة في المجال {100 – 0 }.

```
#include<iostream>
using namespace std;
void main()
{
int i=1, sum=0;
while(i<=100)
{ if(i%2==1)
sum += i ;
i++; }
cout<<"sum="<<sum;
}
```

النتيجة التي تظهر على شاشة الحاسب:

Sum=2525

مثال على جملة الدوران While

```
#include <iostream.h>
Void main ( )
{
   int i  , j ;   1
    while (I < 100) && (J < 100)   2
    {
    statement1;   3
    statement2;
    ……….
    I ++;
    J ++;   4
    }
}
```

```
#include<iostream.h>
void main ()
{
int i =1;   1
while(i<=10)   2
    {
    cout<<i<<endl ;   3
    i++;   4
    }
}
```

- برنامج يقوم بطباعة الأعداد من ١ إلى ١٠

عبارة while التكرارية المتداخلة

- العبارة التكرارية الاعتيادية نحتاجها لتكرار سطر برمجي واحد أو عدة اسطر لغرض ما. أما العبارة التكرارية المتداخلة لا تكرر فقط اسطر برمجية إنما تكرر عبارات while **(أو for, do—while)**.
- العبارة التكرارية الداخلية يمر عليها عند كل عدة للعبارة التكرارية الأم لذلك في كل عدة للام تعد العبارة التكرارية الداخلية جميع عداتها الممكنة حسب شرطها. وشكلها العام:

```
initializing1;
while(boolean_expression1)
{ initializing2;
while(boolean_expression2)
{
statement;
update2;
}
update1; }
```

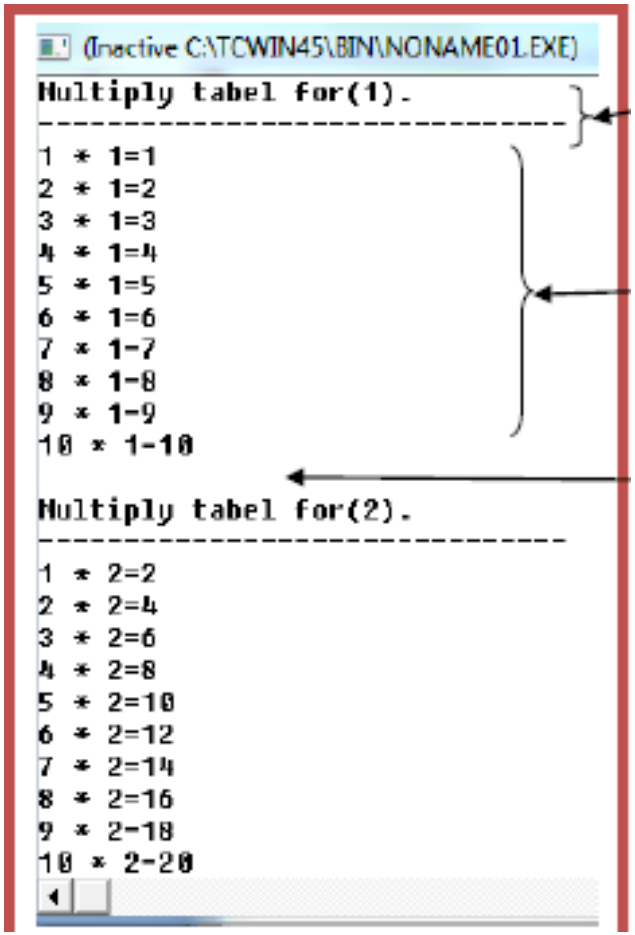
```
i=0;
while( i<5)
{ j=0;
  while(j<5)
  {
   statement
    j++; }
 i++; }
```

- **initializing1,2**: القيمة البدائية المعطاة للمتغير التي منها سيبدأ العد. مثلا (i = 0, J=0)
- **boolean_expression1,2:** شرط التوقف الذي تصبح عنده قيمة المتغير غير محققة لشرط التوقف وسوف يخرج من عبارة while.
- **update1,2:** مقدار الزيادة أو النقصان في قيمة المتغير في كل دورة (i++, J++, i--, J--)
- **statement:** العبارات البرمجية التي سيتم تنفيذها عدة مرات (بحسب العداد في while).

مثال على عبارة while التكرارية المتداخلة

١- برنامج لطباعة جداول ضرب الأعداد من (١ – ١٠)

```
#include<iostream>
using namespace std;
void main()
{   int i=1, j ;
while( i<=10) {
    cout<<"Multiply tabel for("<<i<<“)”;
     cout<<"\n---------------------------\n";
     j=1;
while( j <=10) {
    cout<< j<<" * " <<i<<"="<<i*j<<"\n";
    j++; }
  cout<<"\n";
   i++; }
}
```



جمل الدوران do...While في لغة C++

الجمل الابتدائية
do
عملية واحدة لتكرارها;
while (شرط التكرار)

الجمل الابتدائية
do
{
مجموعة من العمليات لتكرارها
}
while (شرط التكرار)

انتبه في حال كان هناك أكثر من جملة نريد تنفيذها في كل دورة لابد أن نضعهم جميعا بين قوسين

عبارة do .. while التكرارية

- هي عبارة تكرارية تستخدم لتكرار السطور البرمجية المحصورة بين قوسيها لعدد من المرات تكون مشابهة تماما لعبارات التكرار for و while. وتختلف عنهما بأنها تنّفذ العبارات البرمجية الموجودة بين قوسيها أولاً ثم تتحقق من الشرط (تماماً عكس for و while).

```
Initializing;   1
do
{
statement;      2
update;         3
} while ( boolean_expression);   4
```

- لإدخال do .. while في حلقة لانهائية نكتب do .. while(1) فقط

- **initializing**: القيمة البدائية المعطاة للمتغير التي منها سيبدأ العد. مثلا (i = 0)
- **boolean_expression**: شرط التوقف الذي تصبح عنده قيمة المتغير غير محققة لشرط التوقف وسوف يخرج من عبارة while.
- **update**: مقدار الزيادة أو النقصان في قيمة المتغير في كل دورة (i-- , i++)
- **statement**: العبارات البرمجية التي سيتم تنفيذها عدة مرات (بحسب العداد في for).

أمثلة على عبارة do .. while التكرارية

١- برنامج يقوم بطباعة الأعداد من (٠ إلى ٩).

```
#include<iostream>
using namespace std;
void main()
{
int i=0;
do  {
  cout<<i<<"\t";
i++;  }
while(i<10);
}
```

1 2 3 4

النتيجة التي تظهر على شاشة الحاسب:

٠ ١ ٢ ٣ ٤ ٥ ٦ ٧ ٨ ٩

٢- برنامج يقوم بطباعة الأعداد من (٤ إلى ٠).

```
#include<iostream>
using namespace std;
void main()
{
int i=4;
do {
cout<<i<<"\n";
i--;  }
while(i>=0);
}
```

1 2 3 4

النتيجة التي تظهر على شاشة الحاسب:

٤
٣
٢
١
٠

مثال على جملة الدوران do .. While

```
#include <iostream.h>
Void main ( )
{
   int i , j ;   1
    do
    {
    statement1;
    statement2;          2
    ..............
     I ++;
    J ++;                3
    }
    while (I < 100) && (J < 100);   4
}
```

- برنامج يقوم بطباعة الأعداد من ١ إلى ١٠

```
#include<iostream.h>
void main ()
{
int i =1;                1
do
    {
    cout<<i<<endl ;      2
    i++;                 3
    }
while(i<11);             4
}
```

عبارة do .. while التكرارية المتداخلة

- العبارة التكرارية الاعتيادية نحتاجها لتكرار سطر برمجي واحد أو عدة اسطر لغرض ما. أما العبارة التكرارية المتداخلة لا تكرر فقط اسطر برمجية إنما تكرر عبارات while **(أو for, do—while)**.
- العبارة التكرارية الداخلية يمر عليها عند كل عدة للعبارة التكرارية الأم لذلك في كل عدة للام تعد العبارة التكرارية الداخلية جميع عداتها الممكنة حسب شرطها. وشكلها العام:

```
Initializing1;
do
{ Initializing2;
do
  { Statement;  Update2; }
while(boolean_expression2) ;
Update1; }
while(boolean_expression1);
```

- **initializing1,2**: القيمة البدائية المعطاة للمتغير التي منها سيبدأ العد. مثلا (i = 0, J=0)
- **boolean_expression1,2**: شرط التوقف الذي تصبح عنده قيمة المتغير غير محققة لشرط التوقف وسوف يخرج من عبارة do .. while.
- **update1,2**: مقدار الزيادة أو النقصان في قيمة المتغير في كل دورة (i++, J++, i--, J--)
- **statement**: العبارات البرمجية التي سيتم تنفيذها عدة مرات (بحسب العداد في while).

```
i=0;
do
{
j=0;
do
  {
statement;
j++;
  }
While(j<5);
i++;
}
 while( i<5);
```

استكشاف الأخطاء الشائعة Troubleshooting

سنقدم بعض النقاط الهامة مع الأمثلة التي تساعدك على اكتشاف وتجنب بعض الأخطاء الشائعة، والتي تحدث أثناء الترجمة، ويقع فيها معظم المبتدئين في تعلم لغة C++. فمن أجل تجنب تلك الأخطاء الشائعة، عليك تذكر النقاط التالية:

- ✓ لغة C++ حساسة لحالة الحروف.
- ✓ الكلمات الأساسية تكتب بالحروف الصغيرة.
- ✓ يجب الإعلان عن المتغيرات قبل استخدامها.
- ✓ يجب وضع فاصلة منقوطة في نهاية كل عبارة، باستثناء العبارات المركبة والتي لا نضع فيها فاصلة منقوطة يمين القوس الأيمن }.
- ✓ عند استخدام إجراء أو كائن من مكتبة C++ القياسية، يجب أن تضم ملف الرأس الذي يحتويه، كذلك يجب أن تحدد فضاءه باستخدام مُعامِل تحديد المجال (::) أو باستخدام عبارة using. فمثلا cout تتطلب ضم ملف الرأس iostream. وهو معرف في الفضاء std، لذا يجب إما إضافة العبارة التالية:

```
using namespace std;
```

أو استخدام cout بالصيغة:

```
std::cout
```

- ✓ تأكد من استخدام معامل الإدراج >> مع cout ومعامل الاستخلاص << مع cin وليس العكس.
- ✓ استخدم الرموز الصحيحة للمعامِلات العلائقية وغيرها.