جامعة المستقبل
AL MUSTAQBAL UNIVERSITY

Ministry of Higher Education and Scientific Research

Al-Mustaqbal University

Computer Engineering Techniques Department

C/C++ programming language

second year

Prepared By

MSc. Zaid Ibrahim Rasool

2023-2024

الفصل الثالث

ايعازات القرار والتكرار

DECISION AND REPEAT INSTRUCTIONS

تعتبر ايعازات القرار والتكرار من اهم القواعد في لغة البرمجة. نلاحظ ان جميع الامثلة التي مرت بنا سابقا كانت تنفذ بطريقة متسلسلة اي (خطوة بعد خطوة) وهنا يتبادر الذهن السؤال التالي كيف بالامكان ان ننفذ خطوة معينة داخل البرنامج بدون تسلسل او بمعنى اخر كيف يمكن التحكم بتنفيذ تلك الخطوات مثلما يريد المبرمج؟ لذلك توجب علينا دراسة هذه اللايعازات.

3.1 التركيب الشرطي البسيط اذا (if statement)

هنالك العديد من الحالات التي لايمكن لل مبرمج من التنبؤ بها اثناء كتابة البرنامج, فمثلا اذا اردنا كتابة برنامج لايجاد الجذر التربيعي لاعداد صحيحة يتم ادخالها من قبل المستخدم اثناء تنفيذ البرنامج, في هذه الحالة العدد الصحيح هو يجب ان يكون عدد موجب لاننا لانستطيع من ايجاد الجذر التربيعي لعدد سالب, لمعالجة هذه المشكلة من الممكن استخدام عبارة اذا الشرطية فمثلا نقول (اذا كان العدد موجب اوجد الجذر التربيعي لهذه العدد), نلاحظ هنا في هذه العبارة ان ال (compiler) لايفهم معنى كلمة موجب لذا توجب علينا من عملية استبدالها بما يتناسب مع قواعد لغة C++ فنقول (اذا كان العدد اكبر من او يساوي صفر اوجد الجذر التربيعي لهذا العدد).

الصيغة العامة:

if (condition)

Statement;

يكون عمل (if) اعتمادا على الشرط (condition), اذا كان الشرط صحيح فيتم تنفيذ العبارة التي بعده اما في حال عدم تحقق الشرط يهمل الذي بعده, لاحظ البرنامج التالي:

```
#include<iostream.h>
int main ( )
{
 int x;
 cin>>x;
  if (x > 0)
 cout << " The number is Positive" ;
}
```

مخرجات البرنامج:

```
15
The number is Positive
```

نلاحظ من تنفيذ البرنامج اعلاه, عند اخال قيمة مثلا 15 وهي عبارة عن قيمة موجبة فان الشرط داخل البرنامج سوف يتحقق وبالتالي ظهرت لنا العبارة ,
. (The number is positive)
في حال قام المستخدم بادخال قيمة سالبة للبرنامج بدل القيمة الموجبة هنا نستطيع من استخدام التركيب الشرطي مرتين داخل البرنامج ليبح البرنامج السابق كلآتي:

```
#include<iostream.h>
int main ( )
{
 int x;
 cin>>x;
 if (x > 0)
 cout << " The number is Positive" ;
 if (x < 0)
 cout << " The number is Negative" ;
}
```

مخرجات البرنامج:

```
15
The number is Positive
-8
The number is Negative
```

اذا كان المطلوب تنفيذ اكثر من جملة ضمن شرط معين فيجب وضع الجمل المراد تنفيذها ضمن القوسين { } الخاصة بالبداية والنهاية, وبذلك سيكون واضحا بان الافعال المراد تنفيذها عند تحقق الشرط تبدا بعد الامر(}) وتنتهي بالعبارة التي قبل ({) .

الصيغة العامة

{

Statement 1;

Statement 2;

……

Statement n;

}

Example

Write a program to read two real numbers and then sort those numbers ascending.

```
#include<iostream.h>
void main ( )
{
 float a,b,temp; cin>>a>>b;
 if (a>b)
 {
  temp=a;
  a=b;
   b=temp;
 }
 cout << a <<" , "<< b ;
}
```

3.2 التركيب الشرطي اذا..... والا (if…..else)

الصيغة العامة

```
if ( condition)
Statement _1;
 else
Statement _2;
```

اذا كان الشرط (condition) **صحيحا فنفذ الجملة** (Statement _1) **اما اذا كان الشرط غير صحيح فنفذ الجملة** (Statement _2).

Write a program that inserts an integer, and finds out if this is an odd or even number.

```
#include<iostream.h>
int main ( )
{
int x;
cin>>x;
if (x%2==0)
cout << " The number is Even" ;
else
cout << " The number is Odd" ;
}
```

مخرجات البرنامج:

التنفيذ الاول: 8

The number is Even

التنفيذ الثاني: 11

The number is Odd

نلاحظ عند التنفيذ الاول اي عند ادخال قيمة (8), فن الشرط متحقق وبالتالي ظهرت لنا العبارة The number is Even بينما في التنفيذ الثاني اي عند ادخال قيمة (11), فان الشرط غير متحقق ورغم ذلك ظهرت لنا العبارة The number is Odd .

Example

Write a program to read two valid numbers, and then type the variable that has the largest value.

Example

Write a program to calculate the value of y if you know that

$$\mathbf{Y}= \begin{cases} 5\text{-}x^2 & if \quad x >= 0 \\ 2\text{-}x^3 & if \quad x < 0 \end{cases}$$

4 اذا المركبة (Compound if)

من الممكن استخدام (if) بشكل متداخل مع اخرى, في هذه الحالة تسمى ب (المركبة), اي ممكن ان يكون بعد الشرط الذي بعد (if) عبارة (if) اخرى و ممكن ايضا بعد عبارة (else) في هذه الحالة تكون اكثر من عبارة (if) واحدة. **وتاخذ البنية العاملة لجملة اذا ..... والا (if…….else)**

الصيغة العامة :

```
If (condition)
  {
statmenet1;
   }
else
  {
statmenet1;
   }
```

Example

Write a program to insert a student mark if the student's mark is greater than or equal to (90), the estimate (A), if the student's mark is greater than or equal to (80) the estimate (B), or if the student's mark is greater than or equal to (70) the estimate (C), if the student's mark is greater than or equal to (60) the estimate (D), if the student's mark is greater than or equal to (50) the estimate (E), otherwise the estimate (F).

```
#include<iostream.h>
main ()
{
int mark;
char grade;
cin>>mark;
if (mark>=90)
grade='a';
else
if (mark>=80)
grade='b';
else
if (mark>=70)
grade='c';
else
if (mark>=60)
grade='d';
else
if (mark>=50)
grade='e';
else
grade='f';
cout<<grade;
}
```

في المثال اعلاه قمنا بتعريف المتغير mark بانه عدد صحيح, والمتغير grade بانه قيمة نصية و هو التقدير.

مخرجات البرنامج:

3.5 عبارة التكرار (while LOOP)

اسلوب التكرار باستخام حلقة while هو تكرار لمرات غير محددة, حيث يعتمد على تحقق شرط معين لايقاف التكرار.

الصيغة العامة:

While (condition is true) {

Statement 1;

Statement 2;

Statement n;

}

عمل while يختبر الشرط (**condition**) اولا, فاذا كان الشرط صحيحا, نفذ الجملة (Statement), نكرر هذا الاختبار والتنفيذ لغاية ان يصبح الشرط غير صحيح.

كما هو الحال في (if….else) فان (while) يتفذ عبارة واحدة فقط والتي تاتي بعده مباشرة, اما اذا كان هناك اكثر من عبارة واحدة ومطلوب تكرارها باستخدام (while) فيجب ان تحدد بين قوس البداية ({) وبين قوس النهاية (}) لتكون كتلة تنفذ جميعا.

Example

```
#include <iostream.h>
int main ()
{
 int n ;
 cout << "Enter the starting number : " ; cin >> n ;
 while (n>0)
 {
  cout << n << ", " ;
  --n;
 }
```

```
 cout << "FIRE!" ;
}
```

مخرجات البرنامج:

```
Enter the starting number: 8
8, 7, 6, 5, 4, 3, 2, 1, FIRE!
```

نلاحظ من خلال تنفيذ البرنامج اعلاه انه تم تخصيص قيمة 8 للمتغير n, ثم اختبار الشرط (n>0) فاذا كان الشرط صحيح فيتم طباعة القيمة 8 ثم اصبحت القيمة 7 للمتغير n, من خلال مؤثر النقصان n - - ويبقى الشرط صحيح وهكذا تتم عملية التكرار الى ان تصبح القيمة 0 للمتغير n, في هذه الحالة يصبح الشرط غير صحيح وبالتالي تم الخروج من الحلقة نهائيا, والانتقال الى تنفيذ العبارة FIRE! .

نلاحظ البرنامج الاتي:

Example

```
#include<iostream.h>
void main( )
{
int x;
cout<< " Enter number";
cin>> x ;
while (x != 0)
{
cout<< x ;
cin >> x ;
}
}
```

في هذا البرنامج مطلوب ادخال قيم مختلفة و شرط التوقف هنا هو عند ادخال القيمة (0), معنا ذلك ان امر التكرار سيستمر اكثر من مرة وفي كل مرة ستتم عملية فحص الرقم لغرض عملية الطباعة,هذه العملية من الممكن تكرارها 10 مرات او 20 مرة او 2000 مرة او اكثر حسب طبيعة العمل, لذلك هنا تم استخدام (while) لتقليل عمليات التكرار واستخدام مجموعة من الايعازات المتشابهة و فائدة (while) لاختصار البرنامج, ولاكن هذا الامر يحتاج الى شرط

لاجل التوقف, في هذا البرنامج شرط التوقف هو عند ادخال القيمة (0), اي انه يعمل مع الارقام الاخرى, لذلك جعلنا الشرط (x != 0).

هنالك قاعدة مهمة يجب ملاحظتها في كل مرة يتم قراءة قيمة جديدة للمتغير (x) فان القيمة السابقة ستزول وتحل محلها القيمة الجديدة.

Example

Write a program to calculate and print the total numbers (4.0, 4.5, 5.0, 5.5, …… 9.5, 10)

```
#include <iostream.h>
void main ( )
{
   float a=4.0 , sum=0.0 ;
   while (a<=10)
  {
    sum+=a ;
    a+=0.5 ;
  }
 cout << " The Total = " << sum ;
}
```

Example

Write a program that enters a numeric value, and as long as the value entered is a positive value, type * on a new line.

3.6 عبارة التكرار افعل..... بينما (do…..While)

هذا الامر مشابه للحلقة التكرارية (while) حيث يستخدم هذا الامر لتكرارة عبارة او اكثر عدد من المرات, في هذا الامر فان البرنامج سيقوم بتنفيذ العبارات بين do……while على الاقل مرة واحدة و يكون توقف البرنامج بالاعتماد على شرط يوضع بعد (while).

الصيغة العامة:

```
do {
statement 1 ;
statement 2 ;
etc… }
```

while (condition is true) ;

Example

Write a program to insert and print an integer, stop the program when you enter a value of 0.

```
#include <iostream.h>
int main ( )
{
  long n ;
 do
  {
    cout << " Enter number (0 to end): " ;
    cin >> n ;
    cout << " You entered: " <<n<<endl ;
   }
   while (n != 0) ; return 0 ;
}
```

3.7 ايعازز التكرار (for loop)

يقوم هذا الامر بتكرار ايعاز او مجموعة ايعازات لعدد من المرات المحددة مسبقا.

الصيغة العامة:

for (initialization ; condition ; increase)

statement;

يمثل (**initialization**) القيمة الاولية لعداد الحلقة (عداد الحلقة هو عبارة عن متغير), بعد ذلك اختبار الشرط (**condition**) فاذا كان الشرط صحيح نفذ الجملة (**statement**) بعد ذلك نفذ (**increase**) الذي يمثل مقدار الزيادة او مقدار النقصان بمقدار معين في عداد الحلقة.

عبارة (**for**) لها امكانية كبيرة ومرنة وتكون مفيدة بدرجة عالية اثناء تنفيذ

البرنامج و يمكن ان نوجز تنفيذها ب ثلاث خطوات مهمة :

1- تنفيذ العبارة الاولى للامر **for** وهي اسناد قيمة ابتدائية للمتغير الذي سيعمل كعداد.
2- حساب قيمة الشرط (**true or false**).
3- تنفيذ الجزء الثالث من امر التكرار (**for**) والذي يمثل عداديعد عدد مرات التكرار التي حدثت سواء كان العداد للزيادة او للنقصان.

Example

Program to print the word (Zaid C++) ten times

```
#include<iostream.h>
main()
{
cout<<" Zaid C++\n";
cout<<" Zaid C++\n";
                                        cout<<" Zaid C++\n";
cout<<" Zaid C++\n";
cout<<" Zaid C++\n";
cout<<" Zaid C++\n";
cout<<" Zaid C++\n";
cout<<" Zaid C++\n";
cout<<" Zaid C++\n";
return 0;
 }
```

نعيد كتابة هذا البرنامج باستخدام حلقة التكرار (for)

```
#include<iostream>
void main ()
{
int counter;
for ( counter =1; counter<=10 ; counter++)
cout<< " Zaid C++\n";
return 0;
```

```
}
```