



**AUST** الجامعة العربية الخاصة للعلوم والتكنولوجيا  
**Engineering Faculty** كلية الهندسة المعلوماتية  
**IT** قسم تقانة المعلومات

- المقرر: مدخل إلى البرمجة (١)
- المحاضرة : الخامسة

Dr. Mohammed AL-Mohammed



# Chapter 5

جمل التحكم – الجمل الشرطية

**Conditional Statements**

**في لغة C++**

# جمل التحكم 1-:- الجمل الشرطية

هي عبارة عن جمل تتحكم في كيفية تنفيذ خطوات البرنامج وتنقسم إلى نوعين:

١- جمل شرطية Conditional Statements

٢- جمل تكرار (حلقات تكرارية) Loops Statements

الجمل الشرطية: تتحكم في تنفيذ خطوات البرنامج من خلال شرط أو مجموعة شروط ومن هذه الجمل :

# جملة IF والتي تأخذ ثلاثة أشكال أ-

**1- if (condition)  
statement;**

مثال ١

```
if (x==100)  
    cout<<"x is 100";
```

```
2- if(condition)  
{  
    statement 1;  
    statement 2;  
    statement n;  
}
```

وهذا الشكل يعني أنه إذا تحقق الشرط فإن جملة أو مجموعة من الجمل تنفذ

```
if(x==100)
{
cout<<"x is ";
cout<<x;
}
```

ب-

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

وهذا الشكل يعني انه إذا تحقق الشرط فان جملة أو مجموعة من الجمل تنفذ وإذا لم يتحقق الشرط فان جملة

أو مجموعة من الجمل تنفذ أيضاً

```

if(x==100)
    cout<<"x is 100";
else
    cout <<"x is not 100";
    
```

جـ

```

if (condition1)
    statement1;
else if(condition2)
    statement2;
else if(condition3)
    stataement3;
    :
    else
    statement n;
    
```

```

    if(x>0)
        cout<<"x is positive";
    else if (x<0)
        cout<<"x is negative";
    else
        cout<<"x is 0";

```

أمثلة على جملة if:-

١. اكتب برنامج يقوم بإدخال القيمة الصحيحة x وإذا كانت أكبر من أو تساوي ١٠٠ اطبع large value

```

#include<iostream.h>
main( )
{
    int x;
    cout<<"enter x:";
    cin>>x;
    if ( x>=100)
        cout<<"large value";
}

```

٢. اكتب برنامج لإدخال القيمتين الصحيحتين x و y ثم طباعة القيمة الأكبر

```
#include<iostream.h>
main( )
{
    int x,y;
    cout<<"enter x and y \n";
    cin>>x>>y;
    if (x>y)
        cout<<"max ="<<x;
    else
        cout<<"max="<<y;
}
```



٣. اكتب برنامج لإدخال ثلاث قيم صحيحة ثم طباعة القيمة الأكبر  
الطريقة الأولى :

```
#include<iostream.h>
main( )
{
    int x,y,z;
    cout<<"enter x,y,z";
    cin>>x>>y>>z;
    if(x>y && x>z)
        cout<<"max="<<x;
    else if (y>x && y>z)
        cout<<"max="<<y;
    else
        cout<<"max="<<z;
}
```

```
#include<iostream.h>
main()
{
int x,y,z,m;
cout<<"enter x,y,z";
cin>>x>>y>>z;
m=x;
if(y>m)
m=y;
  if (z>m)
m=z;
cout<<"max="<<m;
}
```

٤. اكتب برنامج لإدخال عدد ثم تحديد فيما إذا كان العدد فردي أم زوجي

```
#include<iostream.h>
```

```
main()
```

```
{
```

```
int number;
```

```
cout<<"enter number :";
```

```
cin>>number;
```

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

```
cout<<"the"<<number<<"is even number ";
```

```
else
```

```
cout<<"the"<<number<<"is odd number";
```

```
}
```

٥. أكتب برنامج لإدخال عددين صحيحين وفحص فيما إذا كان الثاني قاسم للأول ام لا .

```
#include<iostream.h>
main ( )
{
    int a , b ;
    cout<<"enter a=";
    cin>>a;
    cout<<"enter b=";
    cin>>b;
    if ( (b!= 0) && (a % b == 0) )
        cout << a << " is divisible by " <<b ;
    else
        cout <<a<<" is not divisible by " << b ;
}
```

## جملة Switch

تستخدم هذه الجملة الشرط المغلق **close condition** أي الشرط الذي يستخدم قيمة محددة (رقمية أو حرفية) ولهذا يمكن القول بأن هذه الجملة تتعامل مع متغير واحد يمكن ان يأخذ في كل مرة قيمة محددة من خلال مجموعة من القيم الصيغة العامة للجملة :

```
switch(variable)
{
case value 1: statement(s);break;
case value 2: statement(s);break;
:
:
case value n: statement(s);break;
default :statement(s);
```

هذه الصيغة تمثل الاحتمالات التي يمكن أن يأخذها المتغير **variable** من **[1-----n]**

كل احتمال يتم وضعه داخل جملة باستخدام العبارة **case** ثم بعد ذلك كتابة الحدث الذي يمكن أن يكون

تعليلة أو مجموعة تعليمات ولا بد أن ينتهي الحدث بواسطة العبارة **break** التي تمثل نهاية الجملة أما

الاحتمال **default** يمثل الحدث الملازم للحالة خارج النطاق من **[1-----n]**

مثال ١ :-

اكتب برنامج لإظهار القائمة التالية :

**Main Menu**

**1-college**

**2-Department**

**3-address**

**Enter your choice[1----3]**

وعلى المستخدم اختيار واحدا من الخيارات الثلاثة بإدخاله رقم الخيار المطلوب  
حيث

(١) يطبع Teachers' college (٢) يطبع computer (٣) يطبع Riyadh

```

#include<iostream.h>
main()
{
int x;
cout<<"\t\t\t Main Menu";
cout<<"\n\t\t 1-college \n\t\t 2-Deparment ";
cout<<"\n\t\t 3-address ";
cout<<"\n\t\t Enter your choice [1-3]\n";
cin>>x;
switch(x)
{
case 1:cout<<"Teachers' college "; break;
case 2:cout<<"Computer"; break;
case 3:cout<<"Riyadh"; break;
default:cout<<"out of range";
}
}
}

```

مثال ٢: أكتب برنامج لإعطاء اسم اليوم من أيام الأسبوع عند ادخال رقمه

```
#include <iostream.h>

main ()
{
    int d ;
    cout << "enter number : " ;
    cin >> d ;
    switch (d )
    {
        case 1 : cout << " Saturday " ; break ;
        case 2 : cout << " Sunday " ; break ;
        case 3 : cout << " Monday " ; break ;
        case 4 : cout << " Tuesday " ; break ;
        case 5 : cout << " Wednesday " ; break ;
        case 6 : cout << " Thursday " ; break ;
        case 7 : cout << " Friday " ; break ;
        default : cout << " The number out of range(1----7) " ;
    }
}
```

مثال ٣: أكتب برنامج لإدخال القيمتين X

وY ثم اجراء العمليات الحسابية

(الجمع، الطرح، الضرب، القسمة، باقي

القسمة)على هاتين القيمتين.



## جمل التحكم - ٢ -

### الجزء الثاني : الحلقات التكرارية

ب- الحلقات التكرارية (جمل التكرار) Iteration Statements

تستخدم الحلقة التكرارية في تكرار تنفيذ تعليمة أو مجموعة من تعليمات أكثر من مرة وهي نوعان :

البرنامج في هذا النوع يتم تحديد عدد مرات التكرار بواسطة رقم صحيح ويبين هذا النوع على مفهوم العداد counter وهو عبارة عن متغير يأخذ قيمة ابتدائية Initial Value تتغير باستمرار بمعدل معين بالزيادة أو النقصان الى ان تصل الى القيمة النهائية Final value ولتصميم هذه البنية نستخدم الجملة for للصيغة العامة لحلقة for

```
for(counter=initial_value ; condition ; step)
```

حيث:

الشرط:- هو عبارة عن جملة تقارن بين قيمة العداد الحالية والقيمة النهائية .

**Step:-** تعبير رياضي يوضح التغير في العداد في كل دورة

الشكل العام للحلقة

```
for(      )  
    statement;
```

أو

```
for(      )  
{  
    Statement 1;  
    .  
    Statement n;  
}
```

امثلة على حلقة for

١. اكتب برنامج لطباعة الارقام من ١ الى ١٠ ومربعاتها

```
#include<iostream.h>
main()
{
    int i;
    for(i=1;i<=10;i++)
        cout<<i<<"\t"<<i*i<<"\n";
}
```

٢. اكتب برنامج لحساب متوسط القيم من 123\_\_\_\_\_477

```
#include<iostream.h>
main()
{
int i,n=0,sum=0;
float avg;
for(i=123;i<=477;i++)
{
n++;
sum+=i;
}
avg=sum/n;
cout<<"average="<<avg;
}
```

٣. اكتب برنامج لحساب مضروب  $n$

حيث :  $n! = 1 \times 2 \times 3 \times 4 \times \dots \times n$

أو  $n! = n \times (n-1) \times (n-2) \times \dots \times 2 \times 1$

أ- الطريقة الاولى :

```
#include<iostream.h>
main()
{
int n,f=1,i;
cout<<"enter n =";
cin>>n;
for (i=1;i<=n;i++)
    f=f*i;
cout<<"factorial="<<f;
}
```

```
#include<iostream.h>
main()
{
int n , f ,i ;
cout<<"Enter n:" ;
cin>>n ;
f=1;
for(i=n ; i>=1 ; i--)
f*=I ;
cout<<"factorial ="<<f;
}
```

٤- أكتب برنامج لحساب  $a^b$  (باستخدام عملية  $*$ )

```
#include<iostream.h>
main()
{
int a , b , z , i ;
cout<<"Enter a , b : " ;
cin>>a>>b;
z=1;
for(i=1 ; i<=b ; i++)
z=z*a ;
cout <<"a^b="z;
}
```

٥- أكتب برنامج لإيجاد جمع المتسلسلة التالية :  
حيث  $n$  يتم إدخالها من قبل المستخدم

```
#include<iostream.h>
main()
{
int n , i , sum=0;
cout<<"enter n:";
cin>>n ;
for (i=1 ; i<=n ; i++)
sum+=1/i
cout<<"sum="<<sum ;
}
```



## ٦. برنامج إيجاد قواسم عدد X

الحل: نختبر الأعداد التي قبل X بحيث إذا كان باقي القسمة عليها يساوي الصفر عندئذ يكون العدد قاسما للعدد X فإذا فرضنا أن العدد  $x=6$  فإن القواسم هي ٢ و ٣

```
# include < iostream . h >
main ()
{
    int x ;
    cout << " enter number : " ;
    cin >> x ;
    for ( int i = 1 ; i < x ; i ++ )
        if ( x % i == 0 )
            cout << i << " \n";
}
```

## ٧. أكتب برنامج لطباعة :

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

باستخدام الحلقات المتداخلة

```
#include<iostream.h>  
  
main()  
{  
    int i , j ;  
    for (i=1 ; i<=5 ; i++){  
        for ( j=1 ; j<=i ; j++){  
            cout<<"* " ;  
        }  
        cout<<"\n";}}}
```

## الحلقات المشروطة (غير المحددة)

هذا النوع غير محدد الدورات وينتهي تنفيذ الحلقة عندما يتحقق شرط معين ، أو بالأصح تستمر الحلقة في التنفيذ طالما الشرط صحيح وتستخدم الجملة **while()** ويمكن أن تؤدي دور الجملة

**.for()**

الصيغة العامة لجملة **while**

```
while (condition)  
{  
statement 1 ;  
statement 2 ;  
  
statement n ;  
}
```

مثال : حوّل جملة : ( `for (i=1 ; i<=10 ; i++` ) إلى جملة `while`

```
i=1 ;  
while (i<=10)  
{  
i++;  
}
```

يتم استخدام `while()` كدالة تكرارية مفتوحة لتعريف متغير يأخذ قيمة ابتدائية معينة ثم بعد ذلك

وضع الشرط طالما قيمة المتغير ثابتة استمر . ويجب إعطاء المستخدم إمكانية تغيير هذه القيمة

بعد كل دورة بواسطة دالة الإدخال

## أمثلة على حلقة while ()

١- أكتب برنامج لطباعة **c++ language** أكثر من مرة :

```
#include<iostream.h>
main()
{
char ok='y' ;
while (ok=='y')
{
cout<<"\n c++ language";
cout<<"\n continue [y / n]" ;
cin>>ok ;
}
}
```

٢- أكتب برنامج لطباعة الأعداد من 1 ----n بشكل تنازلي :

```
#include<iostream.h>
```

```
main()
```

```
{
```

```
int n ;
```

```
cout<<"enter the stating number :" ;
```

```
cin>>n;
```

```
while (n>0)
```

```
{
```

```
cout<<n<<"\n" ;
```

```
-- n ;
```

```
}
```

```
}
```

### ٣- أكتب برنامج لإيجاد متوسط n من القيم الصحيحة المدخلة من قبل المستخدم

```
#include<iostream.h>
main()
{
int n , i , x, sum ;
float avg ;
sum=0 ; i=1 ;
cout<<"enter n:";
cin>>n;
while (i<=n) {
// enter 10th numbers
cout<<"enter x"<<i<<"\n";
cin>>x ;
sum+=x;
i++ ;
}
avg=sum/n ;
cout<<"the average ="<<avg<<"\n";
}
```

## ٢- حلقة : do-while

هذه الجملة مشابهة لجملة while وتختلف عنها في أمرين :

تبدأ بالعملية أولاً ويتم التحقق من الشرط في أسفل الجملة.

لا بد من تنفيذ الجمل الموجودة بين do و while مرة واحدة على الأقل حتى ولو كان الشرط غير متحقق.

الصيغة العامة لها :

```
Do  
{  
Statement 1;  
.  
.  
.  
Statement n ;  
}while (condition) ;
```



\* أمثلة :

١. اكتب برنامج يقوم بإيجاد مربعات الأعداد من ١ إلى ١٠ ؟

```
# include < iostream.h >
```

```
main( )
```

```
{
```

```
int i;
```

```
i=1; // i is a counter
```

```
do
```

```
{
```

```
cout << i*i<<"\n";
```

```
i++;
```

```
}while ( i<=10);
```

```
}
```

أكتب برنامج لحساب حاصل ضرب جميع الأعداد الزوجية من ٢ إلى ١٠ ؟

```
# include < iostream.h >
```

```
main( )
```

```
{
```

```
int i, product;
```

```
product=1;
```

```
i=2;
```

```
do
```

```
{
```

```
product * = i;
```

```
i + = 2;
```

```
} while (i<=10);
```

```
cout<<" product = " << product;
```

```
}
```

### ٣. اعد كتابة الآتي باستخدام جملة do-while

```
a- int i;  
for (i=1;i<=10;i++)  
cout <<"i="<<i<<" \n";
```

الحل/

```
int i;  
i=1;  
do  
{  
cout<< " i="<<i<<"\n";  
i++;  
} while (i<=10);
```

**b- short a;**

a=1;

for ( ; a++<=10;)

cout<<a<<" \n ";

الحل:

short a;

a=1;

do

{

cout<<a<<" \n ";

a++;

} while (a<=100);

## تدريب

١. باستخدام جملة التكرار while اكتب برنامج لإيجاد حاصل جمع  
الآتي :-

$$a- \text{sum } 1 = 1+2+3+4+\dots n$$

$$b- \text{sum } 2 = 1-2+3-4+\dots n$$

٢. اكتب برنامج لإيجاد حاصل جمع مربعات الأعداد الصحيحة الفردية الواقعة بين عددين  
صحيحين يقوم بإدخالهما المستخدم عن طريق لوحة المفاتيح ؟

## الكلمات المحجوزة: هي الكلمات المعروفة مسبقا من قبل المترجم ل C++

والتي كما سبق و قلنا بأنه لا يجوز استخدامها كأسماء للمتغيرات لأنها حجزت مسبقا من قبل المترجم و لكل كلمة من هذه الكلمة معنى خاص و ترمز إلى شيء معين سوف تتعرف إلى معانيهم عند البدء في كتابة كود البرنامج

ملاحظه: هذه الكلمات المحجوزة تكتب في الأحرف الصغيره

Sizeof	near	Static	asm	Double	long
else	do	int	While	new	auto
if	For	This	Void	Delete	Goto
Case	const	try	char	Class	Public
Private	Continue	Extern	struct	inline	float
Static	Virtual	Volatile	Frinde	enum	near
Typedef	soid	Default	inline	Overload	Unsigned
Union	Signed	Pascal	Operator	Switch	Template
Const	Register	Protected	far	Catch	char
	break	Return			