

مدخل إلى الخوارزميات والبرمجة هندسة الميكاترونكس سنة أولى

2023-2024

مدرس المقرر: د. عيسى الغنام

Lecture No.5

Part2 C++ Control Structures

اكتب برنامج يقوم بتبديل قيمتي متحولين. Example:

<pre>1. # include &lt;iostream&gt; 2. using namespace std; 3. 4. 5. int main () 6. { 7. int a=-12,b=50; 8. cout&lt;&lt;a&lt;&lt;" "&lt;&lt;b&lt;&lt;endl; 9. int t=a; 10. a=b;b=t; 11. cout&lt;&lt;a&lt;&lt;" "&lt;&lt;b&lt;&lt;endl; 12. return 0; }</pre>	<pre>1. # include &lt;iostream&gt; 2. using namespace std; 3. 4. 5. int main () 6. { 7. int a=-66,b=56; 8. cout&lt;&lt;a&lt;&lt;" "&lt;&lt;b&lt;&lt;endl; 9. a=a-b; 10. b=b+a; 11. a=b-a; 12. cout&lt;&lt;a&lt;&lt;" "&lt;&lt;b&lt;&lt;endl; 13. return 0;}</pre>
---	---

ما هو خرج البرنامج التالي: Example:

1. #include <iostream>
2. using namespace std;
3. int main()
4. {
5. int a,b,c;
6. cout << "Hello world!\t enter a,b,c:" << endl;
7. cin>>a>>b>>c;
8. if(a>b && a>c)
9. cout << "the greater is a" << endl;
10. if(b>a && b>c)

11. `cout << "the greater is b" << endl;`
12. `if(c>a && c>b)`
13. `cout << "the greater is c" << endl;`
14. `return 0;}`

Example:: اكتب برنامج يقوم بحل المعادلة التالية

$$y = \sqrt{(1 + \log_{10}(x))^3}$$

```
#include <iostream>
#include <math.h>
using namespace std;
int main()
{ double x,y,y1;
  cout << "enter x :";
  cin >> x;
  if(x>0)
  {y = sqrt(pow((1+log(x)),3));
  y1 = sqrt(pow((1+log10(x)),3));
  cout << "y = " << y;
  cout << "\ny1 = " << y1;
  cout << "\nlog(e) = " << log(2.71828);// Euler's number, is a mathematical constant
//approximately equal to 2.71828
  cout << "\nlog10(10) = " << log10(10);}
  else
  cout << "error";
  return 0;
}
```

```
"E:\0.SCIENCE\0.courses\1.programming languages\c++\0 course\byc
enter x :2.7828
y = 2.87833
y1 = 1.73607
log(e) = 0.999999
log10(10) = 1
Process returned 0 (0x0)   execution time : 4.921 s
Press any key to continue.
```

Exmapple:

اكتب برنامج يسمح بإدخال عدد ويطبعه على الشاشة موجبا أيا كانت إشارته باستخدام Ternary Operator .

```
#include <iostream>
using namespace std;
int main ()
{
int y, x;
cin>>x;
y=(x >= 0)? x :-x;
cout<<y;
//cout<< (x >= 0)? x :-x; //not correct. it results true or false
return 0;}
```

## 1-2 C++ Switch Statements

Use the switch statement to select one of many code blocks to be executed.

Syntax

```
switch(expression) {
    case x:
        // code block
        break;
    case y:
        // code block
        break;
    default:
```

```
// code block  
}
```

This is how it works:

- The **switch** expression is evaluated **once**
- The value of the expression is compared with the values of each **case**
- If there is a match, the associated block of code is executed
- The **break** and **default** keywords are optional, and will be described later in this lecture.

The example below uses the weekday number to calculate the weekday name:

Example

```
int day = 4;  
switch (day) {  
  case 1:  
    cout << "Monday";  
    break;  
  case 2:  
    cout << "Tuesday";  
    break;  
  case 3:  
    cout << "Wednesday";  
    break;  
  case 4:  
    cout << "Thursday";  
    break;  
  case 5:  
    cout << "Friday";  
    break;  
  case 6:  
    cout << "Saturday";  
    break;  
  case 7:  
    cout << "Sunday";  
    break;  
}  
// Outputs "Thursday" (day 4)
```

## 2-2 The break Keyword

When C++ reaches a **break** keyword, it breaks out of the **switch** block. This will stop the execution of more code and case testing inside the block. When a match is found, and the job is done, it's time for a

break. There is no need for more testing. A break can save a lot of execution time because it "ignores" the execution of all the rest of the code in the switch block.

If break is ignored, all cases will be executed after first match.

EXAMPLE

اكتب برنامج يطبع على الشاشة welcome إذا أدخلت 1  
أو يطبع على الشاشة hello student إذا أدخلت 2  
أو يطبع على الشاشة introduction to c++ إذا أدخلت 3  
وإلا يطبع على الشاشة the end

```
#include <iostream>
using namespace std;
int main()
{
    int x;
    cin>> x;
    switch (x)
    {
        case 1:cout<<"welcome"<<endl;break;
        case 2:cout<<"hello student"<<endl;break;
        case 3:cout<<"introduction to c++"<<endl;break;
        default:cout<<"the end";
    }
    return 0;
}
```

## Part 3 C++ Loops

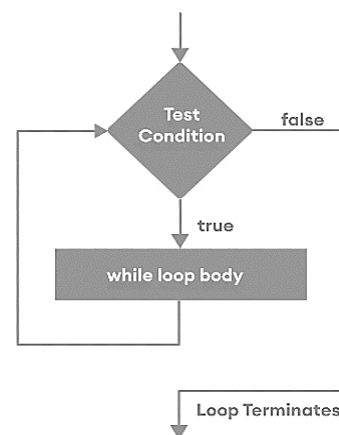
Loops can execute a block of code as long as a specified condition is reached. Loops are handy because they save time, reduce errors, and they make code more readable.

### 1-3 C++ While Loop

The while loop loops through a block of code as long as a specified condition is true:

Syntax:

```
while (condition)
{
    // code block to be
    executed
}
```



In the example below, the code in the loop will run, over and over again, as long as a variable (i) is less than 5:

Example

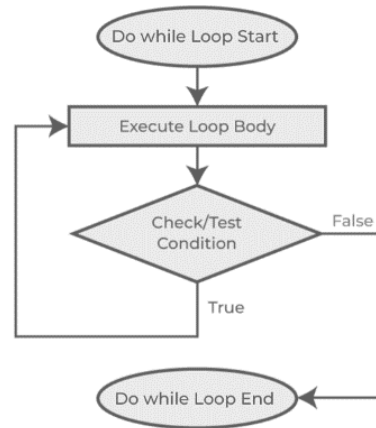
```
int i = 0;
while (i < 5) {
    cout << i << "\n";
    i++;
}
```

### 2-3 The Do/While Loop

The do/while loop is a variant of the while loop. This loop will execute the code block once, before checking if the condition is true, then it will repeat the loop as long as the condition is true.

Syntax

```
do {
    // code block to be
    // executed
}
while (condition);
```



The example below uses a do/while loop. The loop will always be executed at least once, even if the condition is false, because the code block is executed before the condition is tested:

Example

```
int i = 0;
do {
    cout << i << "\n";
    i++;
}
while (i < 5);
```

Exmaple: اكتب برنامج يسمح للمستخدم بإدخال 10 أعداد صحيحة موجبة عن طريق الكي بورد احسب مجموع هذه الأعداد و متوسطها الحسابي .

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    unsigned int num;
```

```
    int sum = 0;
```

```
    float ave;
```

```
    int n = 0;
```

```
cout << "enter ten positive integer numbers " << endl;

while (n<10){

    cin >> num;

    sum = sum + num;

    n++;

}

ave = sum / 10.0;

cout << "sum=" << sum << "\nave=" << ave << endl;

return 0;

}
```

مثال 7 : اكتب برنامج يسمح للمستخدم بإدخال m عدد صحيح موجب عن طريق الكي بورد احسب مجموع هذه الأعداد و متوسطها الحسابي .

```
#include <iostream>

using namespace std;

int main()
{

    unsigned int num;

    int sum = 0;

    float ave;

    int n = 0;

    int m;

    do{cout << "enter positive num of numbers:";
```



```
cin >> m;}

while(m<0);

cout << "enter " << m << " positive integer numbers " << endl;

while (n<m){

    cin >> num;

    sum = sum + num;

    n++;

}

cout.setf(ios::fixed);

cout.setf(ios::showpoint);

cout.precision(3);

ave = (float) sum / m;

cout << "sum=" << sum << "\nave=" << ave << endl;

return 0;

}
```

مثال 8 : اكتب برنامج يسمح للمستخدم بإدخال عدد موجب و يقوم البرنامج بحساب عاملي هذا العدد.

```
#include <iostream>

using namespace std;

int main()

{ int k;

do{cout << "enter positive numbers:";
```

```
cin >> k;}

while (k < 0);

int i=1;

int c=1;

while(i<=k)

{

    c=c*i;

    i++;

}

cout << k << "! = " << c << endl;

return 0;

}
```

مثال 9 : اكتب برنامج يعطي مجموع سلسلة متتالية من الأعداد بدءا من الواحد و حتى عدد يحدده المستخدم.

$$1+2+3+4+5+\dots+n$$

```
#include <iostream>

using namespace std;

int main()

{ int n,sum=0;

    cout << "enter the end number of the sequence greater than zero:";

    cin>>n;

    //do{sum=sum+n;
```

```
// n--;}while(n!=0);  
  
while(n!=0)  
{  
    sum=sum+n;  
    n--;  
}  
  
cout<<sum;  
  
return 0;  
  
}
```

انتهت المحاضرة