Conditional and iterative statements

Jordi Cortadella
Department of Computer Science
Maximum of two numbers

• Write a program that reads two numbers and prints the maximum value of both.

• Example:
  – The maximum of 20 and 38 is 38.
  – The maximum of -3 and -8 is -3.
# Maximum of two numbers

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

// This program reads two numbers and 
// prints the maximum value of both

int main() {
    int x, y;
    cin >> x >> y;
    int m;
    if (x > y) m = x;
    else m = y;
    cout << "The maximum value is "
         << m << endl;
}
```

Conditional statement
# Maximum of two numbers

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

// This program reads two numbers and prints the maximum value of both

int main() {
  int x, y;
  cin >> x >> y;
  int m;
  if (x > y) m = x;
  else m = y;
  cout << "The maximum value is " << m << endl;
}
```

Introduction to Programming © Dept. CS, UPC
#include <iostream>
using namespace std;

// This program reads two numbers and
// prints the maximum value of both

int main() {
    int x, y;
    cin >> x >> y;
    cout << "The maximum value is ";
    if (x > y) cout << x;
    else cout << y;
    cout << endl;
}

Introduction to Programming © Dept. CS, UPC
#include <iostream>
using namespace std;

// This program reads two numbers and
// prints the maximum value of both

int main() {
    int x, y;
    cin >> x >> y;
    if (x < y) x = y;
    cout << x << endl;
}

x < y?
true
false
x = y
Repetitive statements

• Assume the following specification:

**Input:** reads a number \( n > 0 \)

**Output:** prints the sequence 1 2 3 ... \( n \)
(one number per line)

• This specification suggests some algorithm with a *repetitive* procedure.
Print the numbers 1..N

> print_numbers

8
1
2
3
4
5
6
7
8
>
Print the numbers 1…N

// Input: reads a number n > 0
// Output: prints the numbers 1...n (one per line)

int main() {
    int n;
    cin >> n;
    cout << 1 << endl;
    cout << 2 << endl;
    cout << 3 << endl;
    cout << 4 << endl;
    ...
    cout << n << endl;
}

How many?
Print the numbers 1...N

// Input: reads a number n > 0
// Output: prints the numbers 1...n (one per line)

```c
int main() {
    int n;
    cin >> n;
    int i = 1;
    while (i <= n) {
        cout << i << endl;
        i = i + 1;
    }
}
```

<table>
<thead>
<tr>
<th>i</th>
<th>i &lt;= n</th>
<th>Loop body</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>true</td>
<td>cout &lt;&lt; 1 &lt;&lt; endl; i = 1 + 1;</td>
</tr>
<tr>
<td>2</td>
<td>true</td>
<td>cout &lt;&lt; 2 &lt;&lt; endl; i = 2 + 1;</td>
</tr>
<tr>
<td>3</td>
<td>true</td>
<td>cout &lt;&lt; 3 &lt;&lt; endl; i = 3 + 1;</td>
</tr>
<tr>
<td>4</td>
<td>true</td>
<td>cout &lt;&lt; 4 &lt;&lt; endl; i = 4 + 1;</td>
</tr>
<tr>
<td>5</td>
<td>true</td>
<td>cout &lt;&lt; 5 &lt;&lt; endl; i = 5 + 1;</td>
</tr>
<tr>
<td>6</td>
<td>false</td>
<td></td>
</tr>
</tbody>
</table>
The *while* statement

• Syntax:

```plaintext
while ( condition ) statement;
```

(the condition must return *true* or *false*)

• Semantics:
  
  – Similar to the repetition of an *if* statement
  
  – The condition is evaluated:
    
    • If *true*, the statement is executed and the control returns to the *while* statement again.
    
    • If *false*, the while statement terminates.
Multiplication table

• Write a program that reads a number $n$ (between 1 and 9) and prints the multiplication table of $n$:

• Example for $n = 6$:

\[
\begin{align*}
6 \times 1 &= 6 \\
6 \times 2 &= 12 \\
6 \times 3 &= 18 \\
6 \times 4 &= 24 \\
6 \times 5 &= 30 \\
6 \times 6 &= 36 \\
6 \times 7 &= 42 \\
6 \times 8 &= 48 \\
6 \times 9 &= 54
\end{align*}
\]
// Input: reads a number 0 < n < 10
// Output: prints the multiplication table of n

int main() {
    int n;
    cin >> n;
    int i = 1;
    while (i <= 9) { // or also (i < 10)
        // print n x i = n*i
        cout << n << " x " << i << " = " << n*i << endl;
        i = i + 1;
    }
}
Summary

• Conditional statements (*if-else*) are used to take decisions that depend on values of variables.

• The *while* statement is the fundamental instruction to iterate under a condition that determines termination.

• The control of conditional and loop statements is determined by Boolean expressions.