CMP 1001

INTRODUCTION TO PROGRAMMING

PART 4: REPETITIVE STATEMENTS IN C++

by

Assist. Prof. M. Şükrü Kuran

CMP 1001

Section 6: Repetitive statements in C++

WHILE STATEMENT

Repetitive execution in a program

We have seen the sequential statements and conditional statements in C++.

We need a way of writing code that executes some statements repeatedly (i.e., "**repetition statements**"). Statements inside a repetition statement block are instructions that the compiler executes REPEATEDLY while some conditions are satisfied.

```
while (<comparisons>){
    <statements>
}
```

WHILE STATEMENT

Example

Example:
Write a program that prints 100 lines with their line numbers.

Example:
Write a program that computes the average of several numbers entered by the user. The program should read numbers until a negative value is entered.

Example:
Write a program that reads a sequence of characters. Then, counts the lowercase letters in this sequence until * is met.

NOTE: DO NOT put at an ';' at the end of a while statement. This is a VERY COMMON MISTAKE.

FOR STATEMENT

Fixed number of repetitive execution in a program

"**for**" statement is the second type of repetition statements in C++. It works a specific number of steps (i.e., iterations).

```
for (initialization step; comparison step; modification step){
                        <statements>
}
```

FOR STATEMENT

Anatomy of the “for statement”

Same as the comparison inside the “**if**” and “**while**” statements. This step works at the **START** of **EACH ITERATION**.

```
for (initialization step; comparison step; modification step){
                        <statements>
}
```

This statement works at the start of the “**for**” statement **ONCE**. This is usually used to initialize a given variable.

The statement works at the **END** of **EACH ITERATION**.

FOR STATEMENT

Example:
Write a program that prints 100 lines with their line numbers.

Example:
Write a program that computes the average of N numbers entered by the user where N is also specified by the user at the beginning of the program.

Example:
Write a program that controls whether the number entered by the user is perfect or not. Perfect number is the positive number whose sum of its positive divisors except itself is equal to itself. For example 6 and 28 are perfect numbers (6 = 1 + 2 + 3).

NOTE: DO NOT put at an ';' at the end of a for statement. This is a VERY COMMON MISTAKE.

FOR STATEMENT

Relationship between For & While

ANYTHING CAN BE DONE WITH FOR CAN BE DONE WITH WHILE.

```
for (i=0; i<10; i++){
   <statements>
}
```

=

```
i=0;
while (i<10){
   <statements>
   i++;
}
```

NESTED REPETITION STATEMENTS

Similar to the "**if**" statements, "**while**" and "**for**" statements can be written inside other "**while**" and/or "**for**" statements. These repetition statements are called nested repetition statements.

Example:

Write a program that reads two integer values, n and k, calculates the formula below (i.e., F(n)), and prints the result to the screen.

$$F(n) = \prod_{i=0}^{k} (n-i)!$$

NESTED REPETITION STATEMENTS

Example:

Write a program, that draws an isosceles (i.e., ikizkenar) triangle using the '*' character, where line count is specified by the user.

lineCount = 4

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

lineCount = 6

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

NESTED REPETITION STATEMENTS

Example:

Write a program that reads an integer value and computes the sum of the factorial of each digit as an integer.

(e.g., assume that the user enters "572" as the input integer. For each digit of the integer, you will compute the factorial (5! + 7! + 2!). Then you will compute the sum of these factorials: 5! + 7! + 2! = 120 + 5040 + 2 = 5162.)

CONTINUE

Halting a repetition block

"**Continue**" causes the next iteration of the enclosing for or while to begin **IMMEDIATELY**.

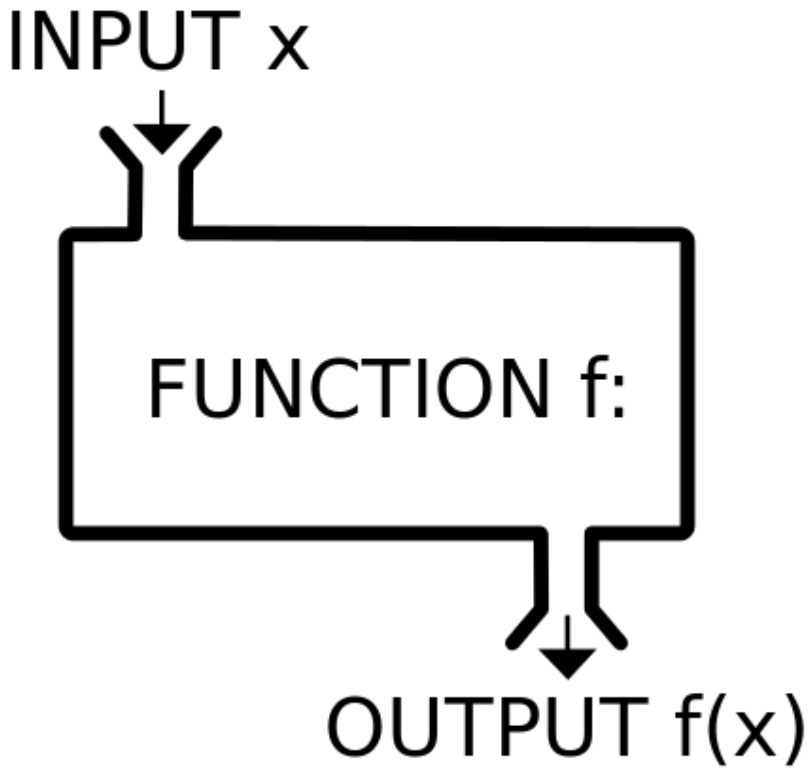
- In "while" loops, the comparison part is executed immediately.
- In "for" loops, the program jumps to the modification step.

Example:

Write a program, that reads a sequence of numbers until the integer value "0" is entered. Calculate the summation of the positive valued numbers in this sequence.

COMING SOON...

Next week on CMP 1001



FUNCTIONS IN C++