CHAPTER 2.2
CONTROL STRUCTURES (ITERATION)

Dr. Shady Yehia Elmashad

Outline

1. C++ Iterative Constructs
2. The for Repetition Structure
3. Examples Using the for Structure
4. The while Repetition Structure
5. Examples Using the while Structure
6. Formulating Algorithms (Counter-Controlled Repetition)
7. Formulating Algorithms with Top-Down, Stepwise Refinement
8. Nested control structures
9. Essentials of Counter-Controlled Repetition
10. The do/while Repetition Structure
11. The break and continue Statements

1. C++ Iterative Constructs

- There are three constructs:
  - while statement
  - for statement
  - do-while statement

2. The for Repetition Structure

The general format when using **for** loops is

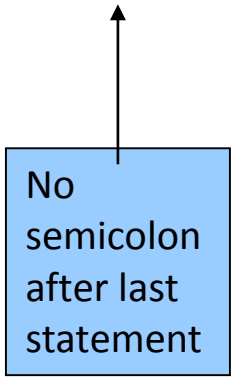
```
for ( initialization;
  LoopContinuationTest; increment ){

    statement(s) }
```

Example:

```
for( int counter = 1; counter <= 10; counter++ ){
   cout << counter << endl;}
```

- Prints the integers from one to ten



2. The for Repetition Structure

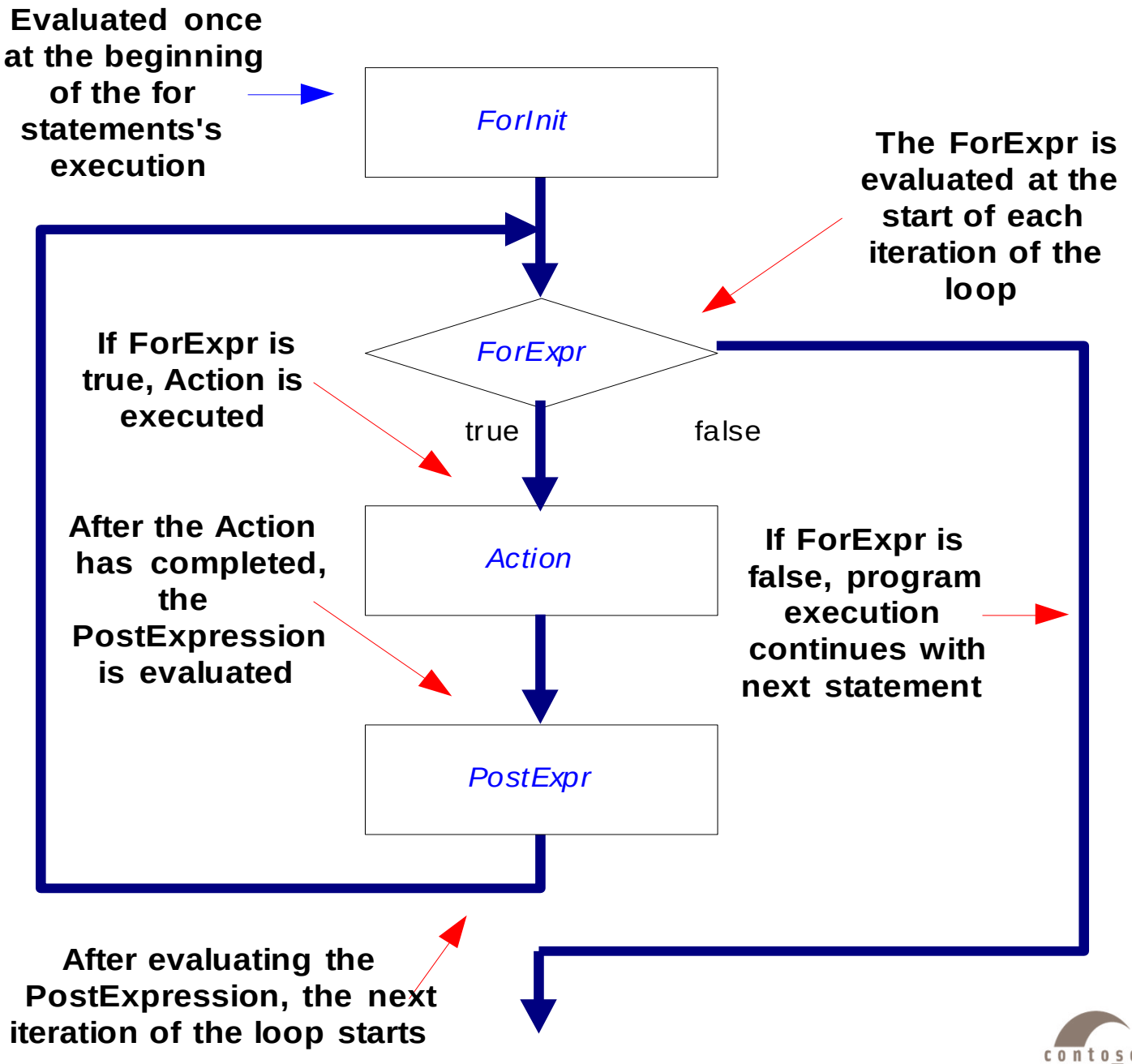
- Syntax

  `for` (*ForInit* ; *ForExpression*; *PostExpression*)

  Action

- Example

```
for (int i = 0; i < 3; ++i) {
  cout << "i is " << i << endl;
}
```


Evaluated once at the beginning of the for statements's execution
ForInit
The ForExpr is evaluated at the start of each iteration of the loop
If ForExpr is true, Action is executed
ForExpr
true
false
After the Action has completed, the PostExpression is evaluated
Action
If ForExpr is false, program execution continues with next statement
PostExpr
After evaluating the PostExpression, the next iteration of the loop starts

2. The for Repetition Structure

- **`For`** loops can usually be rewritten as **`while`** loops:

```
initialization;
while ( loopContinuationTest){
   statement
   increment;
}
```

- Initialization and increment as comma-separated lists

```
for (int i = 0, j = 0;  j + i <= 10; j++, i++)
      cout << j + i << endl;
```

3. Examples Using the `for` Structure

Sum the numbers from 0 to 10

```
#include <iostram.h>
void main ( )
{
int sum = 0 ;
    for ( int i = 0; i < = 10; i++ )
   {
     sum = sum + i ;
     }
cout << " Summation = " << sum ;
}
```

Summation =

3. Examples Using the `for` Structure

Sum the even numbers from 0 to 100

```
#include <iostram.h>
void main ( )
{
int sum = 0 ;
    for ( int i = 0; i < = 100; i+=2 )
   {
     sum = sum + i ;
     }
cout << " Summation = " << sum ;
}
```

Summation =

3. Examples Using the `for` Structure

Sum the odd numbers from 0 to 100

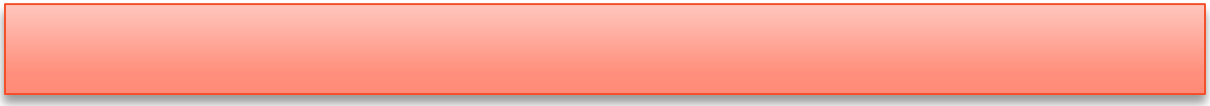
```
#include <iostram.h>
void main ( )
{
int sum = 0 ;
    for ( int i = 1; i < = 100; i+=2 )
   {
     sum = sum + i ;
     }
cout << " Summation = " << sum ;
}
```

Summation =

3. Examples Using the `for` Structure

Printing characters depending on user entry

```
#include <iostram.h>
void main ( )
{
int n ; char  ch;
cout << " Please enter the character: " ;
cin >> ch ;
cout << " Please enter the number of
repetition: " ;
cin >> n ;
      for ( int i = 0; i <  n ; i++ )
      cout <<  ch;
 }
```



4. The while Repetition Structure

Logical expression that determines whether the action is to be executed

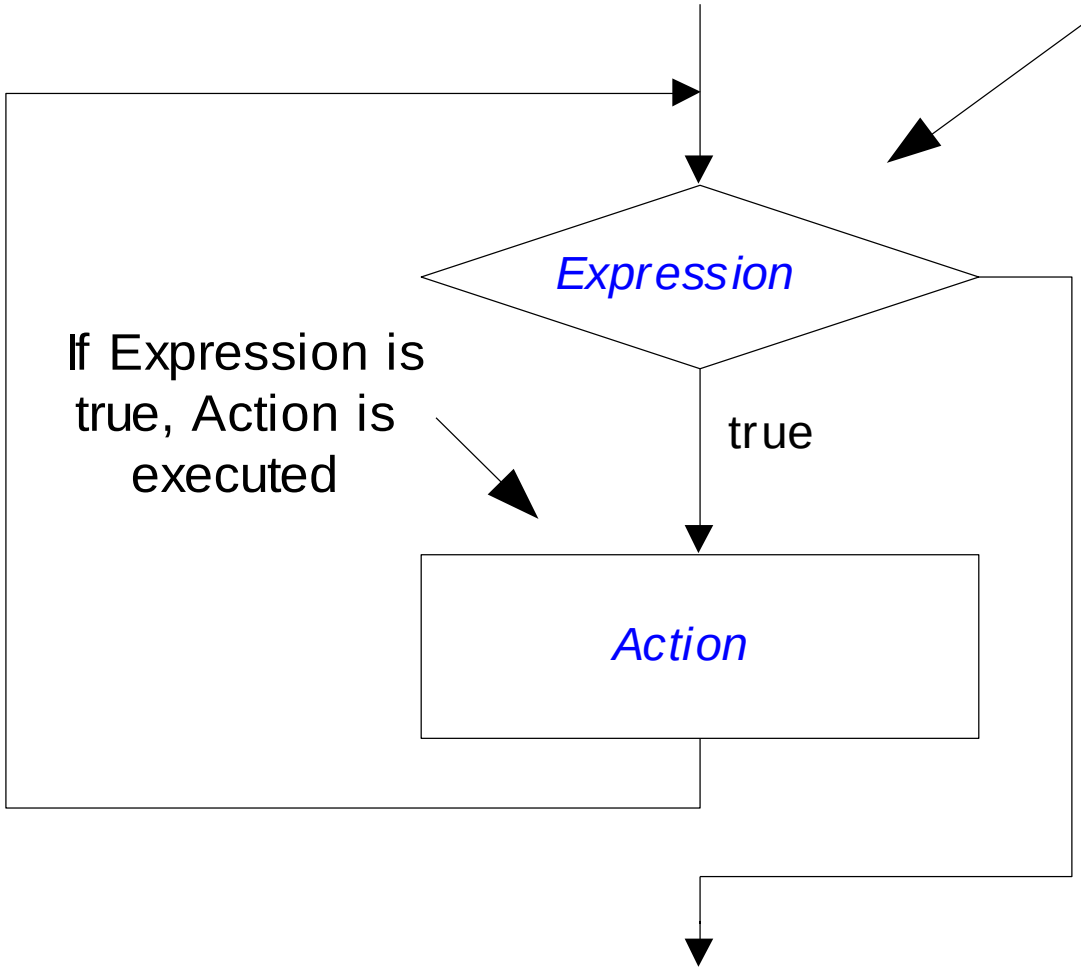
Action to be iteratively performed until logical expression is false

while (*Expression*) *Action*

4. The while Repetition Structure

While Semantics

Expression is evaluated at the start of each iteration of the loop



If Expression is true, Action is executed

If Expression is false, program execution continues with next statement

4. The while Repetition Structure

- Repetition structure
  - Programmer specifies an action to be repeated while some condition remains true
  - Psuedocode

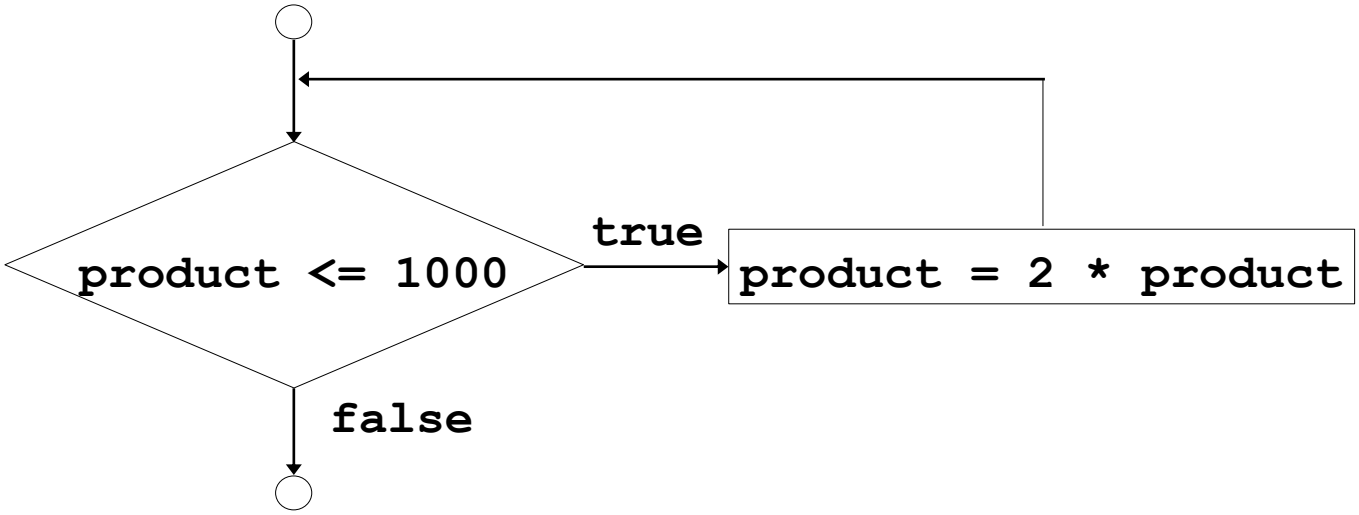
    while there are more items on my shopping list

    Purchase next item and cross it off my list
  - **`while`** loop repeated until condition becomes false.
- Example

```
int product = 2;
while ( product <= 1000 )
   product = 2 * product;
```

4. The while Repetition Structure

- Flowchart of **`while`** loop



5. Examples Using the `while` Structure

Printing characters depending on user entry

```
#include <iostram.h>
void main ( )
{
int n, i = 0 ; char  ch;
cout << " Please enter the character: " ;
cin >> ch ;
cout << " Please enter the number of
repetition: " ;
cin >> n ;
      while ( i <  n )    {
      cout <<  ch ;
      i ++ ;
      }
}
```

5. Examples Using the `while` Structure

The summation of the numbers squared from 0 to 10

```
#include <iostram.h>
void main ( )
{
int sq_sum = 0, x = 0, y ;
     while ( x < =  10 )    {
      y = x * x ;
      sq_sum = sq_sum + y ;
      x ++ ;
      }
cout << “The summation of the
numbers squared from 0 to 10 “ <<
sq_sum ;
}
```

5. Examples Using the `while` Structure

Factorial of a number

```
#include <iostram.h>
void main ( )
{
int n, fact = 1 ;
cout << " Please enter a number " << endl ;
cin >> n ;
      while ( n  > 0 )    {
      fact = fact * n ;
      n -- ;
      }
cout << " The factorial of your number is "
<< fact ;
}
```

8. Nested Control Structures

- Problem:

  A college has a list of test results (1 = pass, 2 = fail) for 10 students. Write a program that analyzes the results. If more than 8 students pass, print "Raise Tuition".

- We can see that
  - The program must process 10 test results. A counter-controlled loop will be used.
  - Two counters can be used—one to count the number of students who passed the exam and one to count the number of students who failed the exam.
  - Each test result is a number—either a 1 or a 2. If the number is not a 1, we assume that it is a 2.
- Top level outline:

  Analyze exam results and decide if tuition should be raised

8. Nested Control Structures

- First Refinement:

  Initialize variables

  Input the ten quiz grades and count passes and failures

  Print a summary of the exam results and decide if tuition should be raised

- Refine

  Initialize variables

  to

  Initialize passes to zero

  Initialize failures to zero

  Initialize student counter to one

8. Nested Control Structures

- Refine

Input the ten quiz grades and count passes and failures

to

While student counter is less than or equal to ten
Input the next exam result
If the student passed
Add one to passes
Else
Add one to failures
Add one to student counter

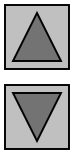
- Refine

Print a summary of the exam results and decide if tuition should be raised

to

Print the number of passes
Print the number of failures
If more than eight students passed
Print "Raise tuition"

```
1  // Fig. 2.11: fig02_11.cpp
2  // Analysis of examination results
3  #include <iostream>
4
5  using std::cout;
6  using std::cin;
7  using std::endl;
8
9  int main()
10 {
11    // initialize variables in declarations
12    int passes = 0,           // number of passes
13        failures = 0,         // number of failures
14        studentCounter = 1,   // student counter
15        result;               // one exam result
16
17    // process 10 students; counter-controlled loop
18    while ( studentCounter <= 10 ) {
19       cout << "Enter result (1=pass,2=fail): ";
20       cin >> result;
21
22       if ( result == 1 )        // if/else nested in while
23          passes = passes + 1;
```

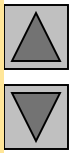


Outline

1. **Initialize variables**

2. **Input data and count passes/failures**

```
24       else
25          failures = failures + 1;
26
27       studentCounter = studentCounter + 1;
28    }
29
30    // termination phase
31    cout << "Passed " << passes << endl;
32    cout << "Failed " << failures << endl;
33
34    if ( passes > 8 )
35       cout << "Raise tuition " << endl;
36
37    return 0;   // successful termination
38 }
```



Outline

3. Print results

Program Output

```
Enter result (1=pass,2=fail): 1
Enter result (1=pass,2=fail): 1
Enter result (1=pass,2=fail): 1
Enter result (1=pass,2=fail): 1
Enter result (1=pass,2=fail): 2
Enter result (1=pass,2=fail): 1
Enter result (1=pass,2=fail): 1
Enter result (1=pass,2=fail): 1
Enter result (1=pass,2=fail): 1
Enter result (1=pass,2=fail): 1
Passed 9
Failed 1
Raise tuition
```

8. Nested Control Structures

Accept 10 numbers from the user & print the max. one

```
#include <iostram.h>
void main ( )
{
int num, largest = 0 ;
   for ( int i = 0; i < 10; i ++ )  {
   cout << " Enter a number: " ;
   cin >> num ;
                  if ( num > largest) {
                  largest = num ;
                  }
    }
cout << " The largest number is " << largest
<< endl ;
}
```