

Introduction to Computers Lab First Year (2017– 2018)

Lab 11

Agenda

- Problem of today
- Introduction
 - While Loop
 - Do...While Loop
 - For Loop
- Examples
 - Loops
 - Break and continue
- Debugging in Visual Studio
- Debugging Example

Problem

- Implement a program that displays numbers from 1 to 10.

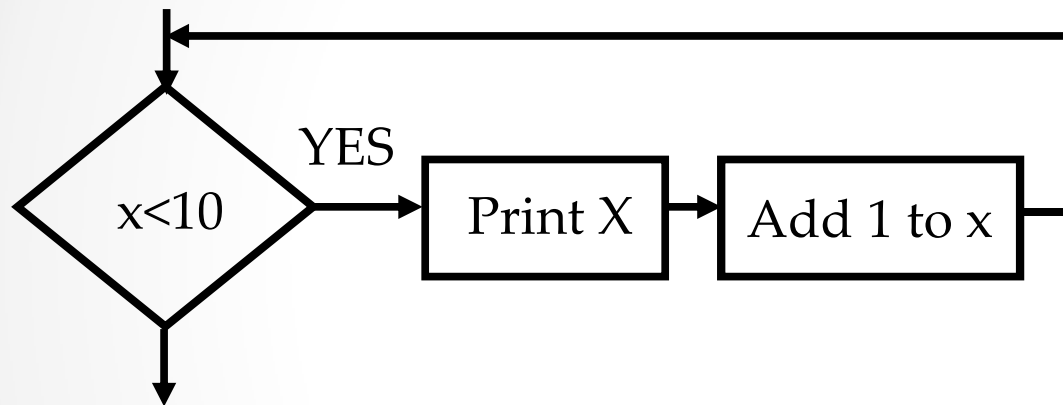


Solution

```
#include <iostream>
using namespace std;
int main()
{
    cout<<"1"<<endl;
    cout<<"2"<<endl;
    cout<<"3"<<endl;
    cout<<"4"<<endl;
    cout<<"5"<<endl;
    cout<<"6"<<endl;
    cout<<"7"<<endl;
    cout<<"8"<<endl;
    cout<<"9"<<endl;
    cout<<"10"<<endl;
}
```



Repetition in flow-charts



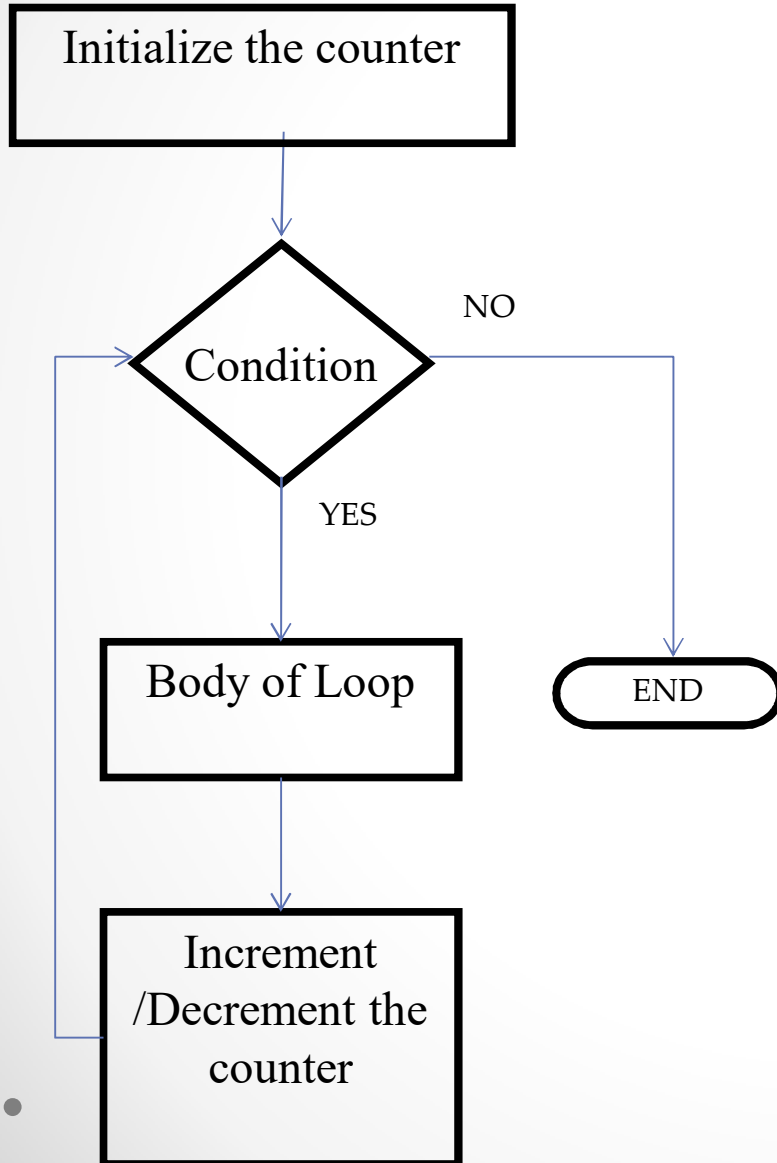
Introduction

- In most software, the **statements** in the program **may need to repeat for many times**.
- **Loop** is a control structure that repeats a group of steps in a program.
 - **Loop body** stands for the repeated statements.
- There are **three C++ loop control statements**:
 - **while**, **for**, and **do-while**.

Any loop contains the following:

- 1- Initialization.
- 2- Condition Checking.
- 3- Increment/Decrement the counter.

While Loop (Flowchart)

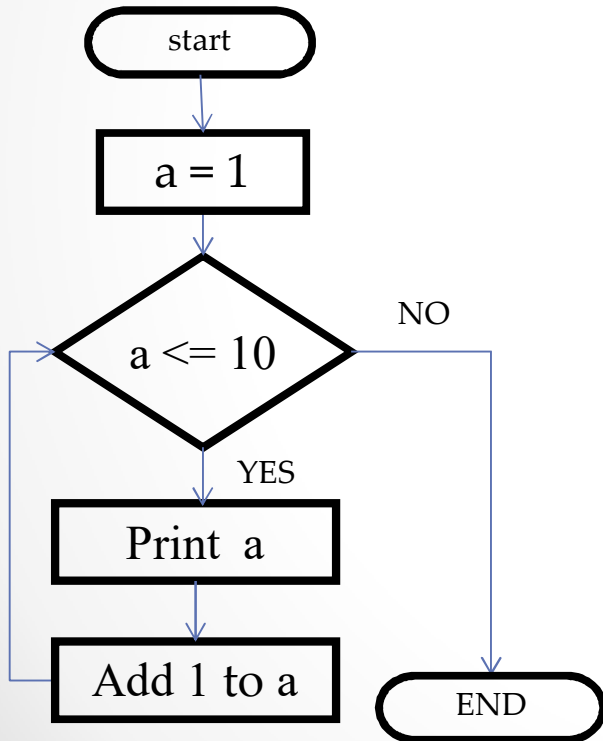


```
while (condition)
{
    statements;
}
```


Example (While Loop)

Implement a program that prints from 1 to 10.

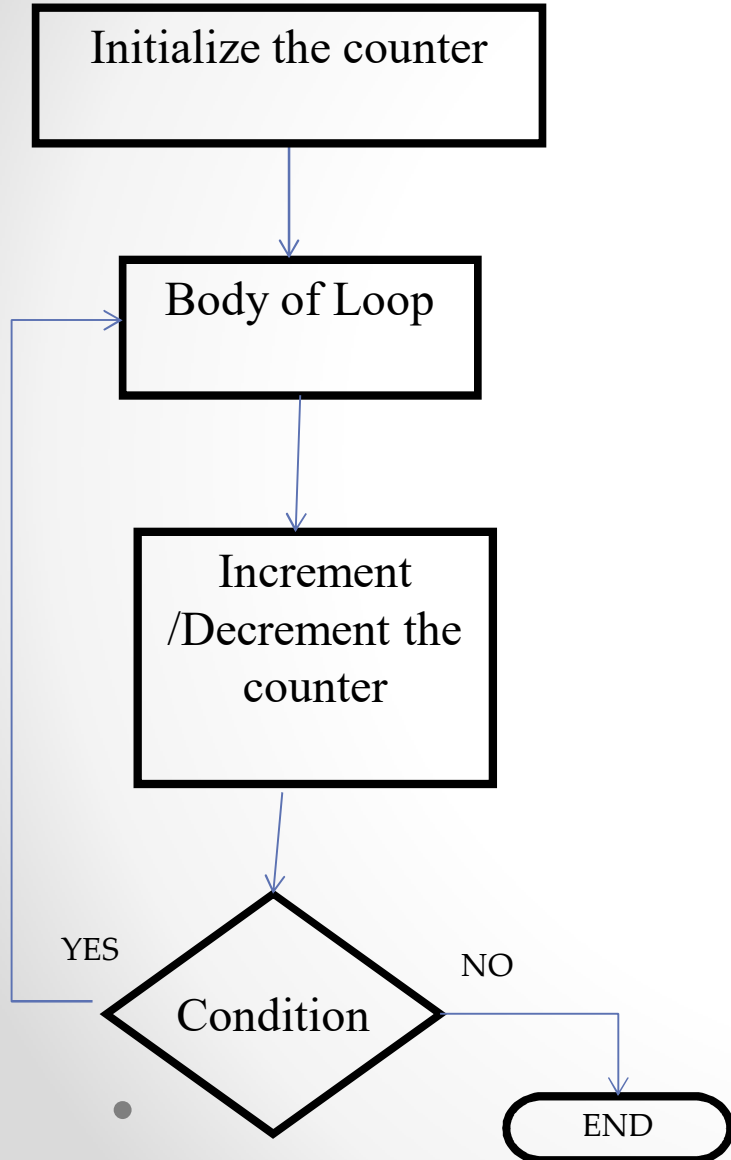
Flowchart



C++ Code

```
int a = 1;
While (a <= 10)
{
    cout << a;
    a++;
}
```

Do...While Loop (Flowchart)

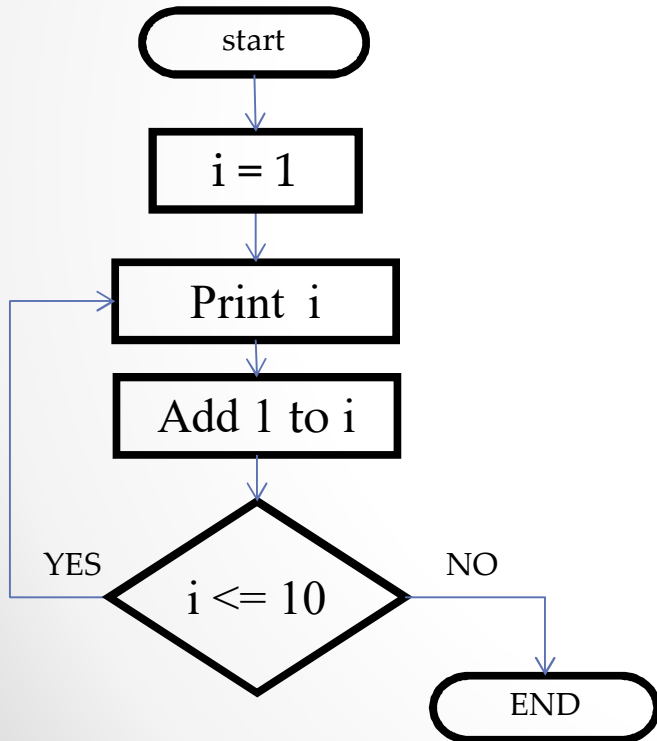


```
do
{
    set of statements;
} while(condition);
```

Example (Do...While Loop)

Implement a program that prints from 1 to 10.

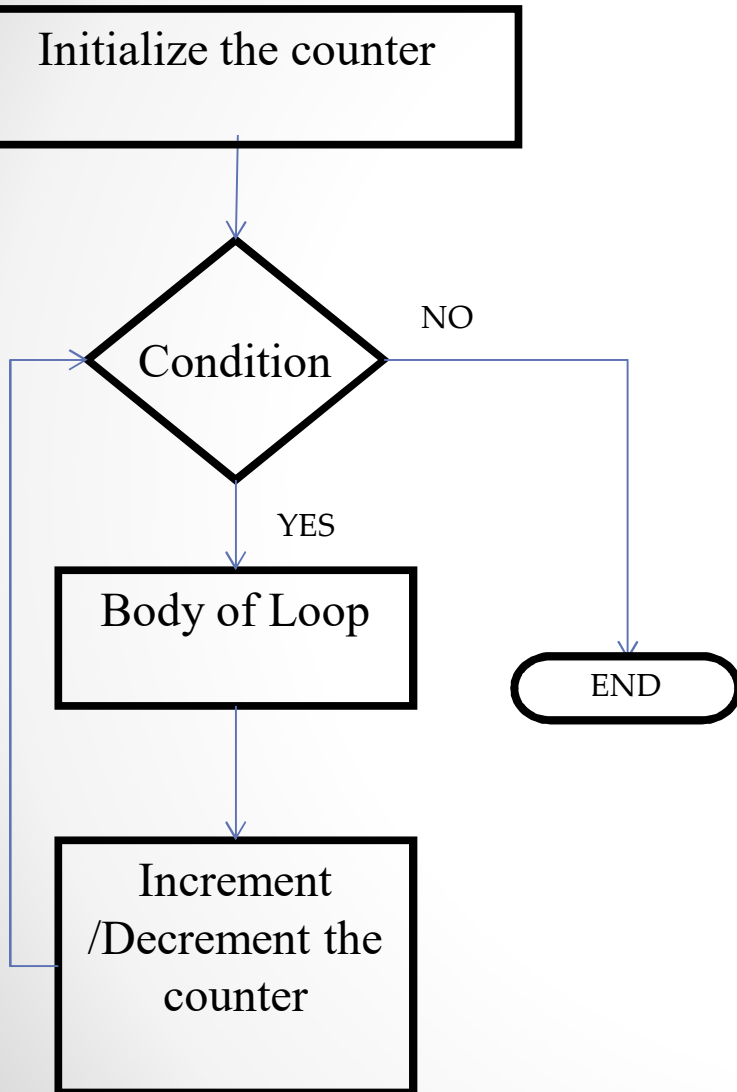
Flowchart



C++ Code

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

For Loop (flowchart)

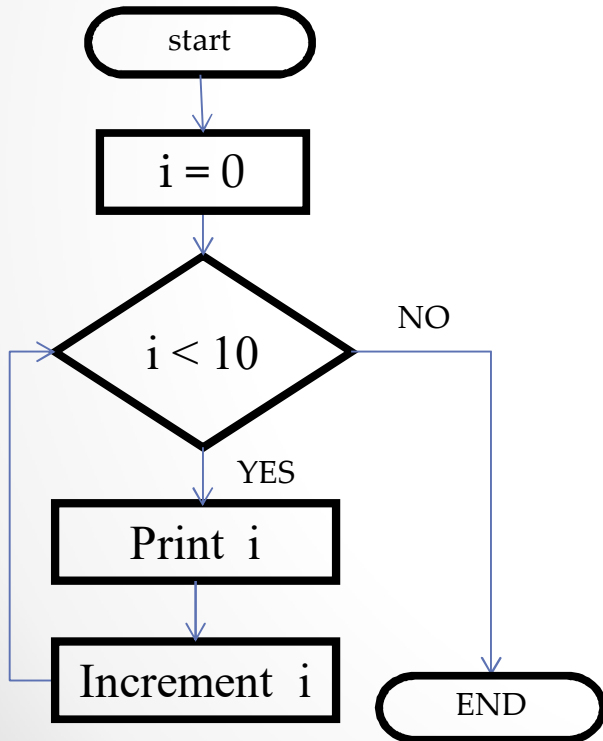


```
for(initialization;  
    condition checking;  
    increment)  
{  
    set of statements;  
}
```

Example (For Loop)

Implement a program that prints **from 0 to 9**.

Flowchart



C++ Code

```
for(int i=0 ; i<10 ; i++)  
{  
    cout<< i << endl;  
}
```

Problem (1)

Write a C++ program that calculate the square of n given numbers.

Enter a number: 3

1 -> 1

2 -> 4

3 -> 9

Solution (using For-Loop)

```
#include <iostream>
using namespace std;
int main()
{
    int c,num;
    int res;
    cout<< "please enter how many number you want to calculate their square \n \n";
    cin>>c;
    for (int i=0;i<c;i++)
    {
        cout<<"Enter the number to square:";
        cin>>num;
        res = num*num;
        cout<<"->"<<res<<endl;
    }

    return 0;
}
```

Solution 2(using While-Loop)

```
#include <iostream>
using namespace std;
int main()
{
    int c,num;
    int res;
    cout<< "please enter how many number you want to calculate their square \n \n";
    cin>>c;
    int i=0;
    while(i<c)
    {
        cout<<"Enter the number to square:";
        cin>>num;
        res = num*num;
        cout<<"->"<<res<<endl;
        i++;
    }

    return 0;
}
```


Solution 3(using Do-While-Loop)

```
#include <iostream>
using namespace std;
int main()
{
    int c,num;
    int res;
    cout<< "please enter how many number you want to calculate their square \n \n";
    cin>>c;
    int i=0;
    do
    {
        cout<<"Enter the number to square:";
        cin>>num;
        res = num*num;
        cout<<"->"<<res<<endl;
        i++;
    } while(i<c);

    return 0;
}
```

Problem (2)

- Write a temperature-conversion program that converts from Fahrenheit to Celsius.

$$T_f = \frac{9.0}{5.0} T_c + 32$$

Then ask the user if he/she needs more operations or not.

Solution

```
#include <iostream>
using namespace std;
int main()
{
    int num;
    float res;
    char ch;
    do
    {
        cout<<"Please enter the temperature in Celsius \n";
        cin>>num;

        res = (9.0/5.0) * num + 32;
        cout<< "Temperature in Fahrenheit = "<<res<<endl;

        cout<<"Press y to do more : ";
        cin>>ch;
    }while(ch=='y');

    return 0;
}
```

Problem (3)

- There are 9870 people in a town whose population increases by 10% each year.
- Implement a C++ program that determines how many years it would take for the population to exceed 30,000.

Solution

```
#include <iostream>
using namespace std;
int main()
{
    int people_number = 9870, years = 0;

    while(people_number<=30000)
    {
        /*this line means: people_number=people_number + people_number*0.1
        people_number += (people_number * 0.1);
        years++;

    }

    cout<<years<<endl;

    return 0;
}
```

Break and continue

- Break

- The **break** statement provides way for terminating the loop to terminate early.

- Continue

- The **continue** statement provides a convenient way to jump back to the top of a loop earlier than normal, which can be used to bypass the remainder of the loop for an iteration

Problem (4) (use break)

- Write a program that accepts numbers from the user and counts the positive and negative numbers. **The results should be displayed when the user enters 0.** A sample run of the program should be like:

```
Enter numbers (0 to end): -1
```

```
4
```

```
-2
```

```
44
```

```
-5
```

```
0
```

```
You entered 2 positive numbers and 3 negative  
ones.
```

Solution

```
#include <iostream>
using namespace std;
int main()
{
    int positive = 0, negative = 0;
    int num;
    while(true)
    {
        cin>> num;

        if(num>0)
            positive++;
        else if(num<0)
            negative++;
        else
            break;
    }
    cout<<"positive count = "<<positive<<endl;
    cout<<"negative count = "<<negative<<endl;
    return 0;
}
```


Solution – Another Solution

```
#include <iostream>
using namespace std;
int main()
{
    int positive = 0, negative = 0;
    int num;
    cout<<" Enter the number.";
    cin>>num;
    while(num!=0)
    {
        if(num>0)
            positive++;
        else if(num<0)
            negative++;
        cout<<" Enter the number.";
        cin>>num;
    }
    cout<<"positive count = "<<positive<<endl;
    cout<<"negative count = "<<negative<<endl;
    return 0;
}
```

Problem (5)

- Write C++ program that allows the user to type characters. If the user hits enter, the program should display the number of entered characters.

Solution

```
#include <iostream>
using namespace std;
int main()
{
    int counter = 0;
    char ch;
    while(true)
    {
        ch = getchar();           //cin>>ch;
        if(ch=='\n')
            break;
        counter++;
    }
    cout<<counter<<endl;
    return 0;
}
```

Problem (6) (use continue)

- Write C++ program that prints all of the numbers from 0 to 19 **that are not divisible by 4**.

Solution

```
#include <iostream>
using namespace std;
int main()
{
    for (int i=0; i < 20; i++)
    {
        // if the number is divisible by 4, skip this iteration
        if ((i % 4) == 0)
            continue;

        cout << i << endl;
    }
    return 0;
}
```

Problem (7)

- Write C++ Program that calculate the factorial for number that is entered by the user.

Note:

The factorial of a number is calculated by the given equation.

$$n! = n * (n - 1) ... 2 * 1$$

Solution

```
#include <iostream>
using namespace std;
int main()
{
    int x;
    int finalx = 1;

    cout<<"Please Enter a number to calculate it's factorial :";
    cin>>x;

    if (x==1)
        cout<<"The factorial of"<< x <<"=" << finalx <<endl;

    else
    {
        for (int i = 1; i <= x; i++)
        {
            finalx = finalx * i;
        }
        cout<<"The factorial of "<< x <<" = " << finalx <<endl;
    }
    return 0;
}
```

Problem 8

- Write a program that repeatedly collects positive integers from the user, stopping when the user enters a negative number or zero. After that, output the product of all positive entries.
- A sample run should appear on the screen like the text below.

```
Enter a number: 3
Enter a number: 10
Enter a number: 2
Enter a number: -13
The product of all your positive numbers is 60.
```


Solution

```
#include <iostream>
using namespace std;
int main( )
{
    int x = 1;
    int product = 1;
    while ( x > 0 )
    {
        product *= x;
        cout << "Enter a number: ";
        cin >> x;
    }
    cout << "The product of all your positive numbers
    is " << product << endl;
    return 0;
}
```

Another Solution: using break

```
#include <iostream>
using namespace std;
int main( )
{
    int x = 1, product = 1;
    while (true)
    {
        cout << "Enter a number: ";
        cin >> x;
        if (x <= 0)
            break;
        product *= x;
    }
    cout << "The product of all your positive numbers is " <<
    product << endl;
    return 0;
}
```

Problem 9

- Write a program that accepts from the user a list of positive integers and displays the number of even and odd values in the list. The program accepts numbers from the user until he enters a negative value.
- A sample run of the previous program is given below:

Enter the numbers:

1

6

3

4

8

-1

The number of even values = 3

The number of odd values = 2

Solution

```
#include <iostream>
using namespace std;
int main()
{
    int x;
    int odd=0, even=0;
    cout<<"Enter positive numbers (negative number to terminate):"<<endl;
    while(true)
    {
        cin>>x;
        if(x<0)
            break;
        else
        {
            if(x%2==0)
                even++;
            else
                odd++;
        }
    }
    cout<<"Even Numbers= "<< even <<endl;
    cout<<"Odd numbers= "<< odd <<endl;
    return 0;
}
```

Problem 10

- Write a program that asks the user to enter an integer and reports all divisors in ascending order.
- An example is shown below, where the user entered 30.

Enter a number: 30

The divisors are: 1 2 3 5 6 10 15 30

Solution

```
#include <iostream>
using namespace std;
int main()
{
    cout << "Enter a number: ";
    int num;
    cin >> num;
    cout << "The divisors are: ";
    for (int i=1; i<=num; i++)
    {
        if (num%i == 0)
            cout << i << " ";
    }
    return 0;
}
```

Problem 11

- Write a program that accepts numbers from the user and then tells the maximum two numbers of them. The results should be displayed when the user enters a negative number.
- A sample run of the program should be like:

Enter numbers (less than 0 to end): 5

33

7

53

52

14

12

45

-1

The maximum two numbers are 53 and 52.

Solution

```
int main()
{
    int x, max_Num1=0, max_Num2=0;
    cout<<"Enter numbers (less than 0 to end): "<<endl;
    while (true)
    {
        cin>> x;
        if(x<0)
            break;
        else
        {
            if(x<max_Num1)
            {
                if(x>max_Num2)
                    max_Num2=x;
            }
            else
            {
                max_Num2=max_Num1;
                max_Num1=x;
            }
        }
    }
    cout<<"Max 1= "<<max_Num1<<endl;
    cout<<"Max 2= "<<max_Num2<<endl;
    return 0;
}
```


Debugging

...

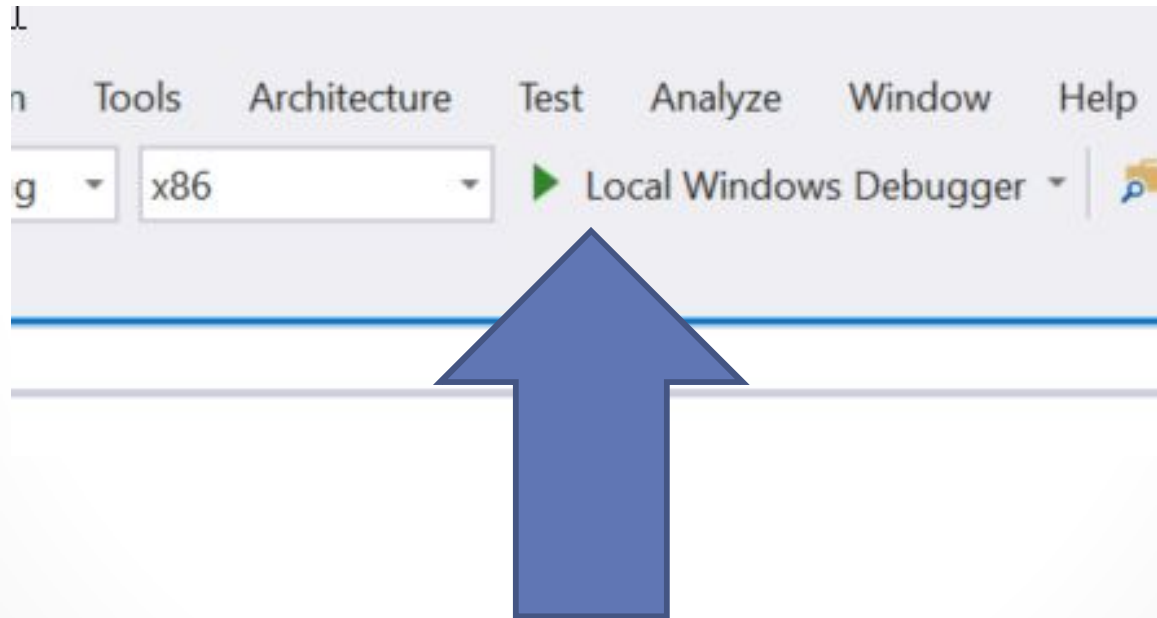
Debugging

- **Debugging** is the process of finding and resolving of defects that prevent correct operation of computer software or a system.

Debugging In Visual Studio

- Run your code in Debug Mode

OR Press F5

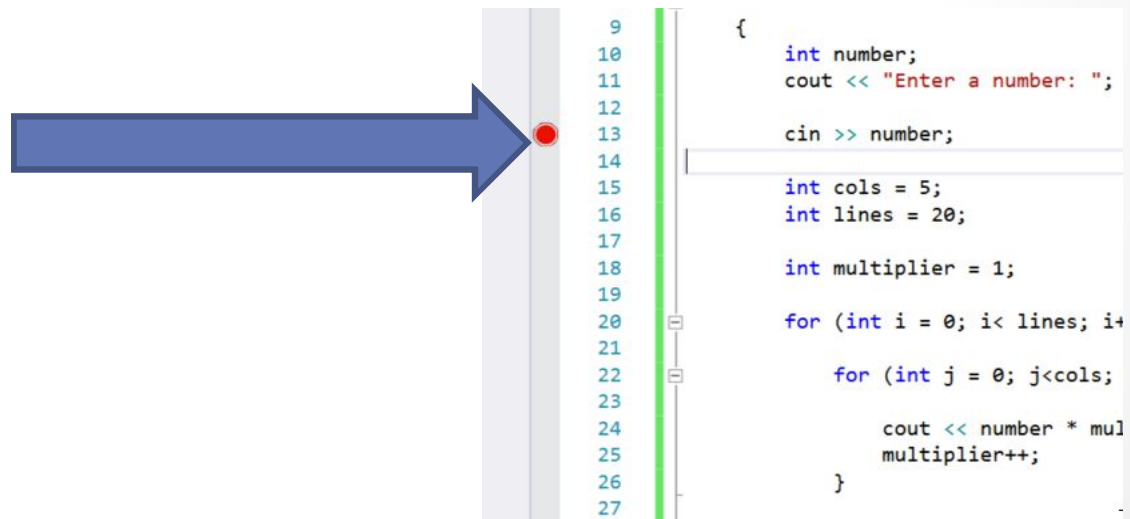


Breakpoints

- A **breakpoint** is an intentional stopping or pausing place in a program, put in place for debugging purposes. It is also sometimes simply referred to as a **pause**.

Setting Breakpoints

1. In a source window, right-click a line of executable code where you want to set a breakpoint.
2. On the shortcut menu, click Breakpoint, and then click Insert Breakpoint.



OR using the mouse; click on the margin of the line you to set a breakpoint

Tracing Code in Debug Mode

In Debug mode you can



Stop Debugging

- Return to the normal mode

Tracing Code in Debug Mode

In Debug mode you can



Restart Debugging

- Stop and Start Debugging

Tracing Code in Debug Mode

In Debug mode you can



Go to Next Statement (Step Over)

- Steps over the current statement to the next one

Tracing Code in Debug Mode

In Debug mode you can



Others used in a later scope

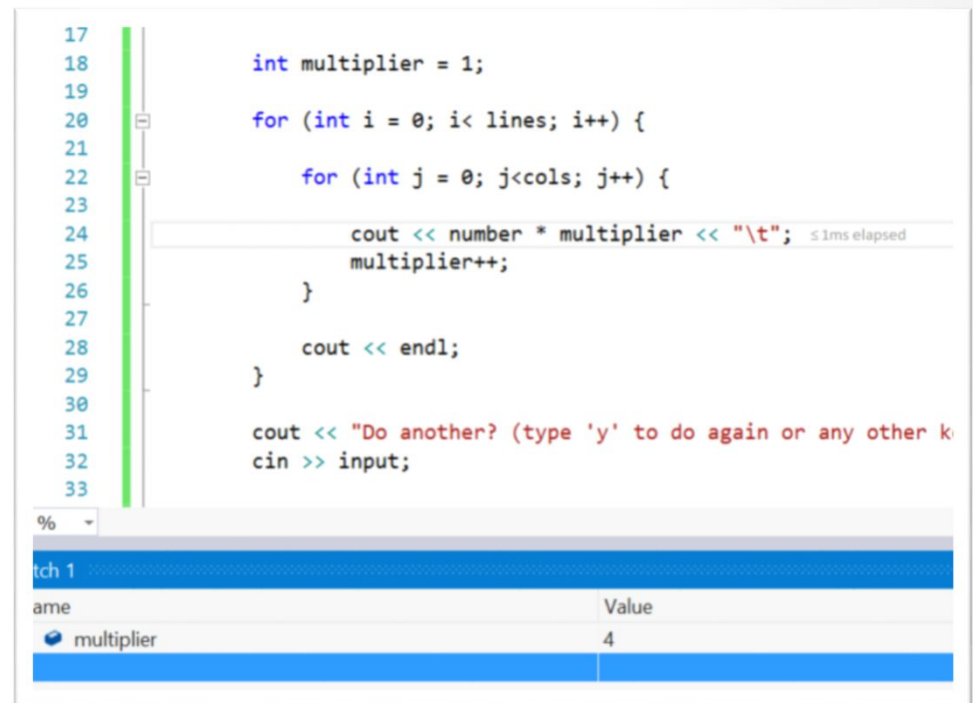
- Other functionalities not relevant now

Tracing Value of Variable

- Use a **Watch** to trace the value of a variable while debugging.

1. Debug Menu >
Windows > Watch >
Watch 1

2. Type the variable
name you want to
trace



The screenshot shows a C++ IDE with a code editor and a Watch window. The code editor displays a program that calculates the product of two numbers, 'number' and 'multiplier', and prints the result. The Watch window shows the variable 'multiplier' with a value of 4.

```
17  
18     int multiplier = 1;  
19  
20     for (int i = 0; i < lines; i++) {  
21  
22         for (int j = 0; j < cols; j++) {  
23  
24             cout << number * multiplier << "\t";  
25             multiplier++;  
26         }  
27  
28         cout << endl;  
29     }  
30  
31     cout << "Do another? (type 'y' to do again or any other k  
32     cin >> input;  
33
```

Watch 1

Name	Value
multiplier	4

Debugging Example

...

Finding the Factorial of a Number

Debugging Example

- Write a program to calculate the factorial of a number provided by the user.

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

int main() {

    int number;
    int factorial = 1;

    cout << "Enter a number : ";

    cin >> number;

    for (int i = 0; i < number; i++)
    {
        factorial = factorial * i;
    }

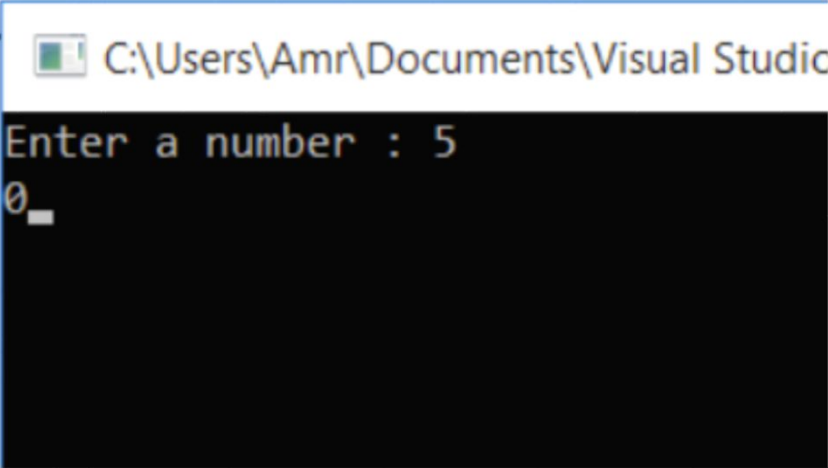
    cout << factorial;

    return 0;
}
```

Debugging Example – cont.

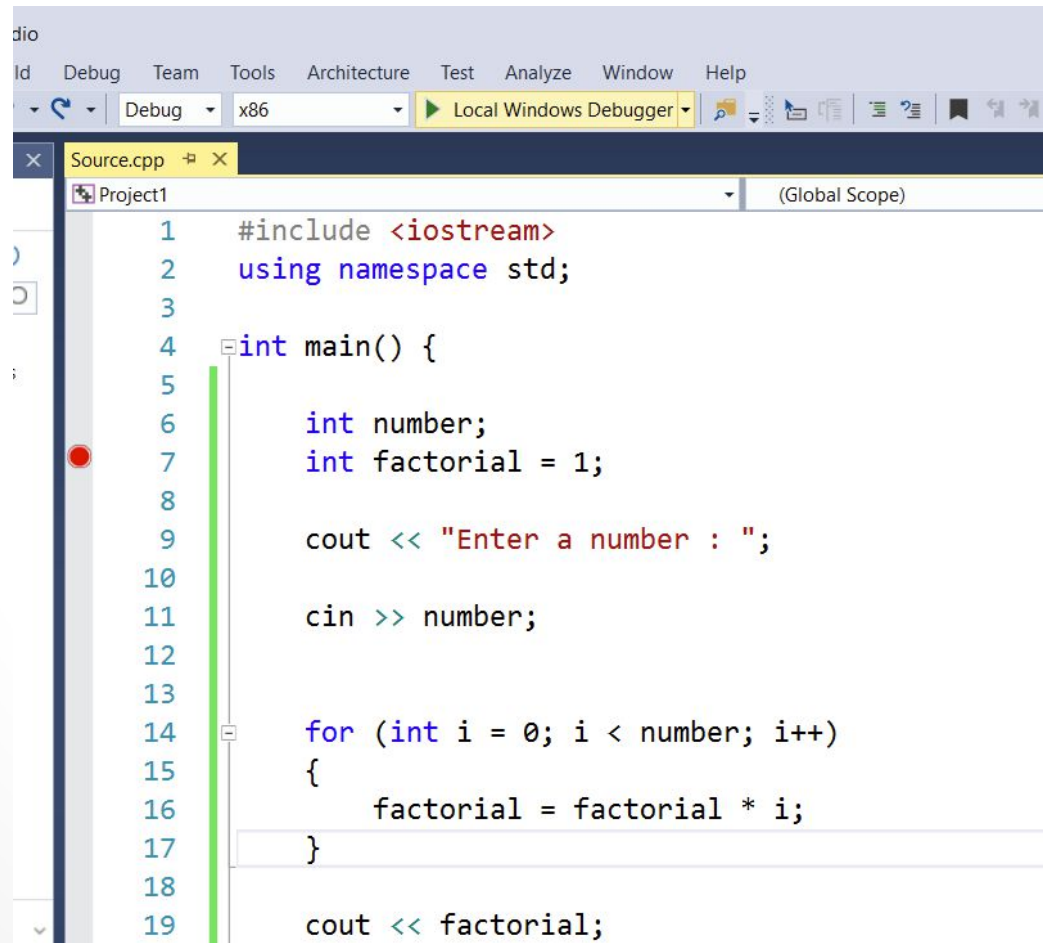
- Wrong answer!

```
int number;  
int factorial = 1;  
  
cout << "Enter a number : ";  
cin >> number;  
  
for (int i = 1; i <= number; i++)  
    factorial = factorial * i;  
  
cout << "Factorial of " << number << " is " << factorial << endl;
```



Debugging Example – cont.

- Set a Breakpoint and start debugging



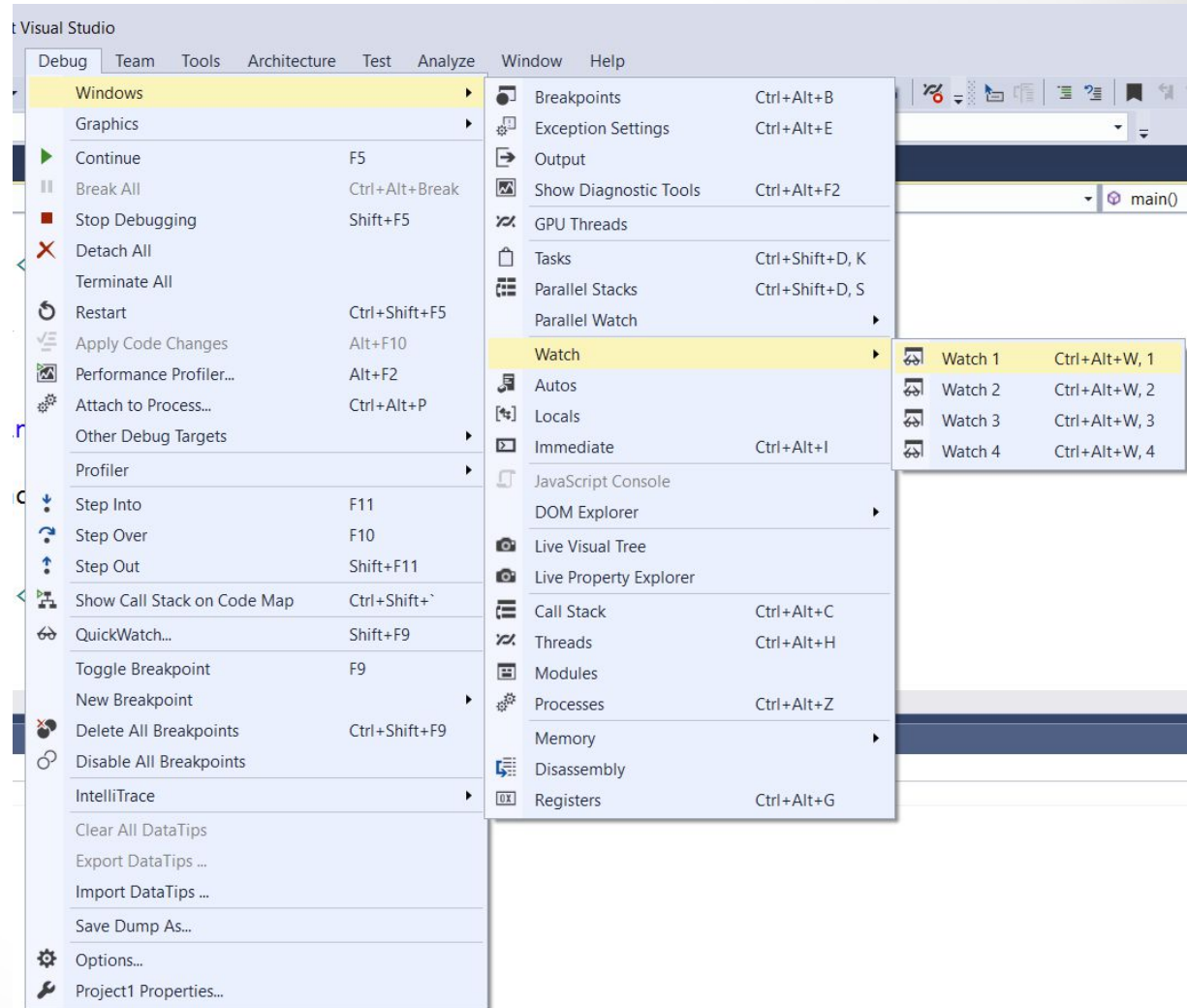
The screenshot shows the Visual Studio IDE with a C++ source file named 'Source.cpp' open. The file is part of 'Project1' and is in the '(Global Scope)'. The code is as follows:

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5
6      int number;
7      int factorial = 1;
8
9      cout << "Enter a number : ";
10
11     cin >> number;
12
13
14     for (int i = 0; i < number; i++)
15     {
16         factorial = factorial * i;
17     }
18
19     cout << factorial;
```

A red dot, representing a breakpoint, is set on line 7, which is the declaration of the 'factorial' variable. The 'Local Windows Debugger' is selected in the toolbar, and the 'Debug' menu is active. A green vertical line is visible in the left margin, indicating the current execution position.

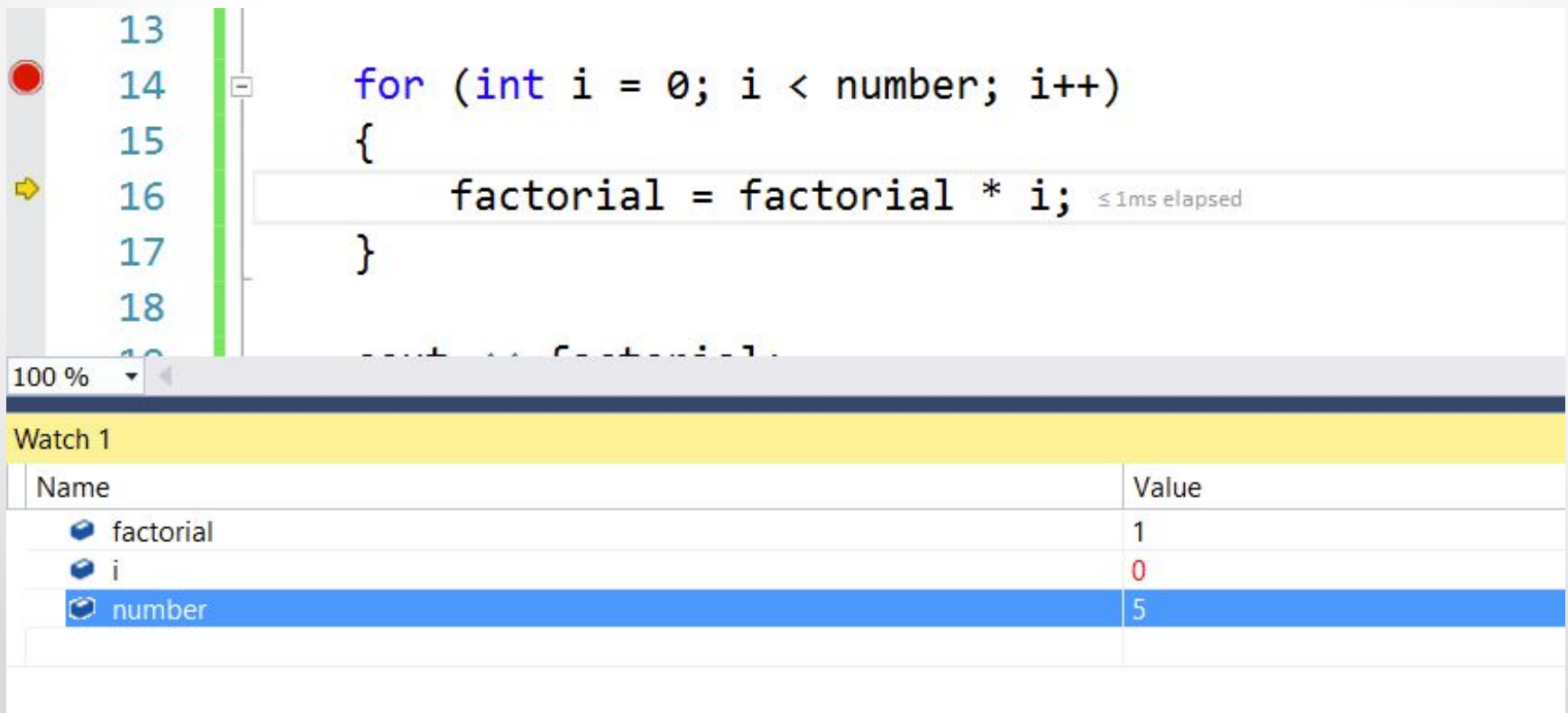
Debugging Example – cont.

- Create a Watch and add all the variables you want to trace.



Debugging Example – cont.

- Step Over the code and observe the values in the Watch Window 



The screenshot shows a debugger interface. On the left, a list of line numbers (13-19) is visible. A red stop button is at the top left, and a yellow arrow points to line 16. The code snippet is as follows:

```
13  
14 for (int i = 0; i < number; i++)  
15 {  
16     factorial = factorial * i; ≤ 1ms elapsed  
17 }  
18  
19
```

Below the code, the Watch Window is displayed with the title "Watch 1". It contains a table with the following data:

Name	Value
factorial	1
i	0
number	5

Debugging Example – cont.

- Find the problem

```
for (int i = 0; i < number; i++)  
{  
    factorial = factorial * i;  
}
```

`int i = 0`

`i` can not start with 0;

(Multiplying by zero makes a wrong solution)

Debugging Example – cont.

- Fix and Restart Debugging

```
5  
6     int number;  
7     int factorial = 1;  
8  
9     cout << "Enter a number : ";  
10  
11     cin >> number;  
12  
13  
14     for (int i = 1; i < number; i++)  
15     {  
16         factorial = factorial * i;  
17     }  
18  
19     cout << "Factorial : ";
```

100 %

Watch 1

Name	Value
------	-------

- Wrong Answer Factorial 4 = 24 not 6

Debugging Example – cont.

- Trace the variable values in the Watch window
- What is the problem?**
The loop finishes before multiplying the last number

```
for (int i = 1; i < number; i++)
```

The screenshot shows a C++ IDE with a program to calculate a factorial. The code is as follows:

```
6  int number;  
7  int factorial = 1;  
8  
9  cout << "Enter a number : ";  
10  
11 cin >> number;  
12  
13  
14  for (int i = 1; i < number; i++)  
15  {  
16      factorial = factorial * i;  
17  }  
18  
19  cout << factorial;
```

A red dot indicates a breakpoint is set at line 14. The execution has stopped at line 19, with a status bar indicating "≤ 1ms elapsed". Below the code editor, the "Watch 1" window is open, displaying the following table:

Name	Value
factorial	6
i	3
number	4

- The loop condition must be less than or equal (\leq)
not less than ($<$)

Debugging Example – cont.

- Fix and Retry

```
process: [7260] Project1.exe
source.cpp
Project1 (Global Scope)
9      cout << "Enter a number : ";
10
11      cin >> number;
12
13
14      for (int i = 1; i <= number; i++)
15      {
16          factorial = factorial * i;
17      }
18
19      cout << factorial;
20
21
22      cin >> factorial;
```

C:\Users\Amr\Documents\Visual Studio 2010\Projects\Project1\Project1.exe

Enter a number : 4

24

- Correct Answer

Thank You 😊