

Session: 4

C# Programming Constructs

- ◆ Explain selection constructs
- ◆ Describe loop constructs
- ◆ Explain jump statements in C#



- ◆ allows you to execute a block of statements after evaluating the specified logical condition.

Syntax

```
if (condition)
{
    // one or more statements;
}
```

where,

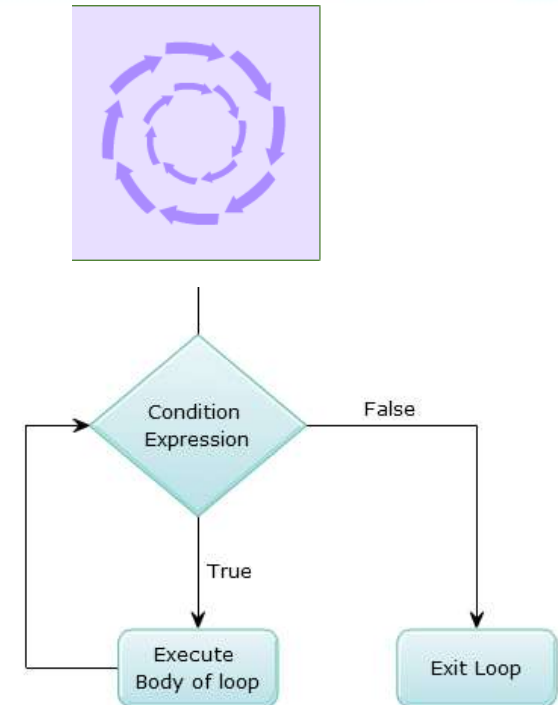
- ◆ `condition`: Is the boolean expression.
- ◆ `statements`: Are set of executable instructions executed when the boolean expression returns true.

- ◆ A program is difficult to comprehend when there are too many **if** statements representing multiple selection constructs.
- ◆ To avoid that, in certain cases, the **switch...case** can be used.
- ◆ The **switch...case** statement is used when a variable needs to be compared against different values.

```
int day = 5;
switch (day) {
    case 1: Console.WriteLine("Sunday"); break;
    case 2: Console.WriteLine("Monday"); break;
    case 3: Console.WriteLine("Tuesday"); break;
    default:
        Console.WriteLine("Enter a number between 1 to 7");
        break;
}
```

Loop Constructs

- ◆ allow to execute a single statement or a block of statements repetitively.
- ◆ contain a condition that identifies the number of times a specific block will be executed.
- ◆ If the condition is not specified, the loop continues infinitely : infinite loop.
- ◆ The loop constructs are also referred to as iteration statements.
- ◆ C# supports four types of loop constructs:
 - ◆ The `while` loop
 - ◆ The `do..while` loop
 - ◆ The `for` loop
 - ◆ The `foreach` loop



- ◆ The **while** loop is used to execute a block of code repetitively as long as the condition of the loop remains true.

Syntax

```
while (condition)  
{  
    // one or more statements;  
}
```

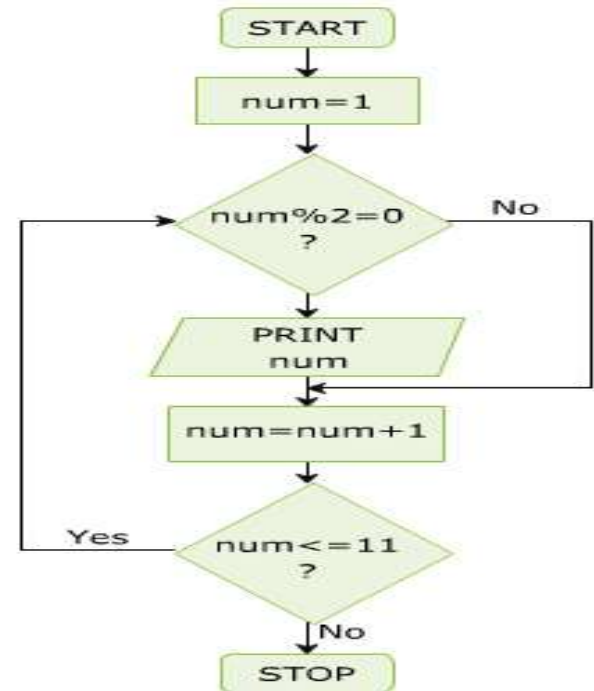
where:

- ◆ `condition`: Specifies the boolean expression.

- ◆ The **do-while** loop is similar to the **while** loop; however, it is always executed at least once without the condition being checked.

Syntax

```
do
{
    // one or more statements;
} while (condition);
```



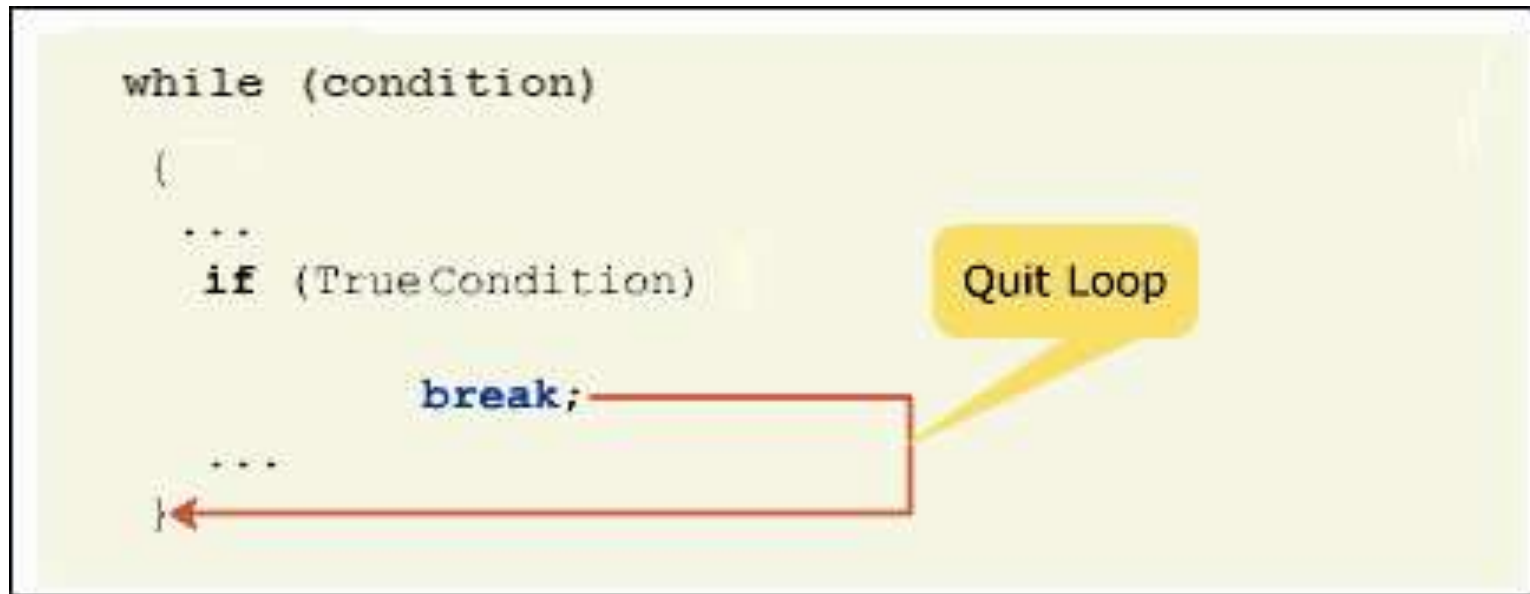
- ◆ is similar to the **while** statement in its function.
- ◆ The statements within the body of the loop are executed as long as the condition is true.

Syntax

```
for (init; condition; increment/decrement)
{
    // one or more statements;
}
```

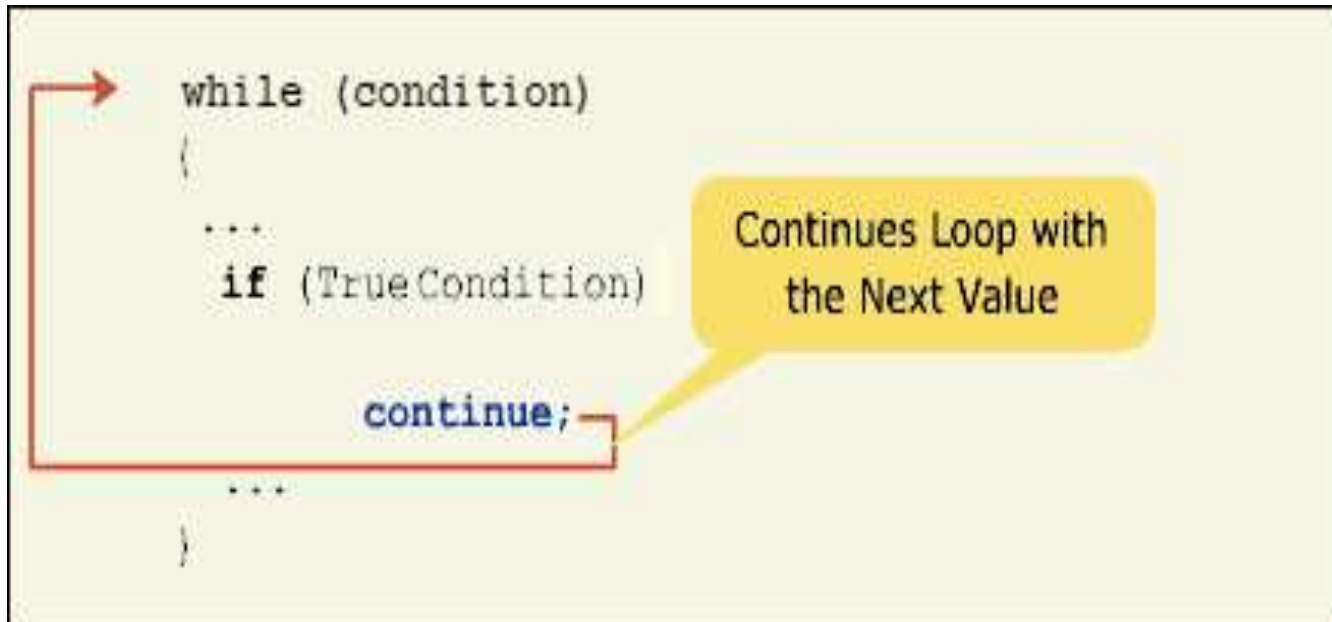
```
int num;
Console.WriteLine("Even Numbers");
for (num = 1; num <= 11; num++) {
    if ((num % 2) == 0) {
        Console.WriteLine(num);
    }
}
```


- ◆ is used in the selection and loop constructs.
- ◆ is most widely used in the **switch...case**, and in the **for** and **while** loops.
- ◆ In loops, it is used to exit the loop without testing the loop condition.



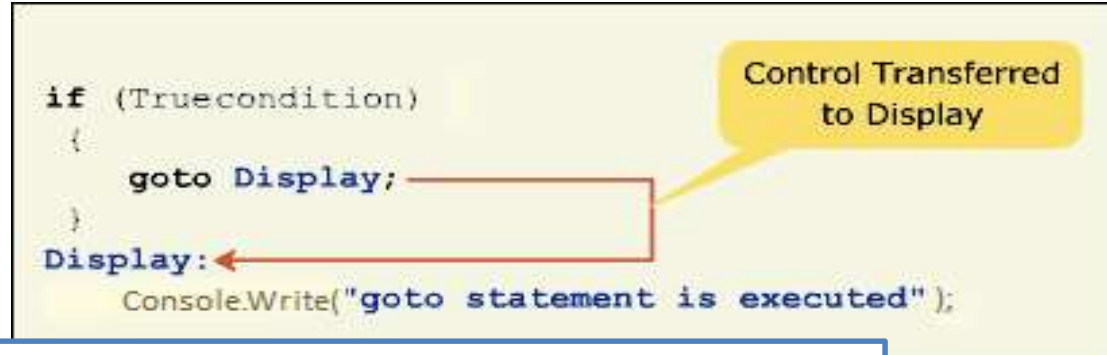
The `continue` Statement

- ◆ is most widely used in the loop constructs
 - ◆ is used to end the current iteration and transfer the program control back to the beginning of the loop.
- The statements of the loop following the **continue** statement are ignored in the current iteration.



The goto Statement

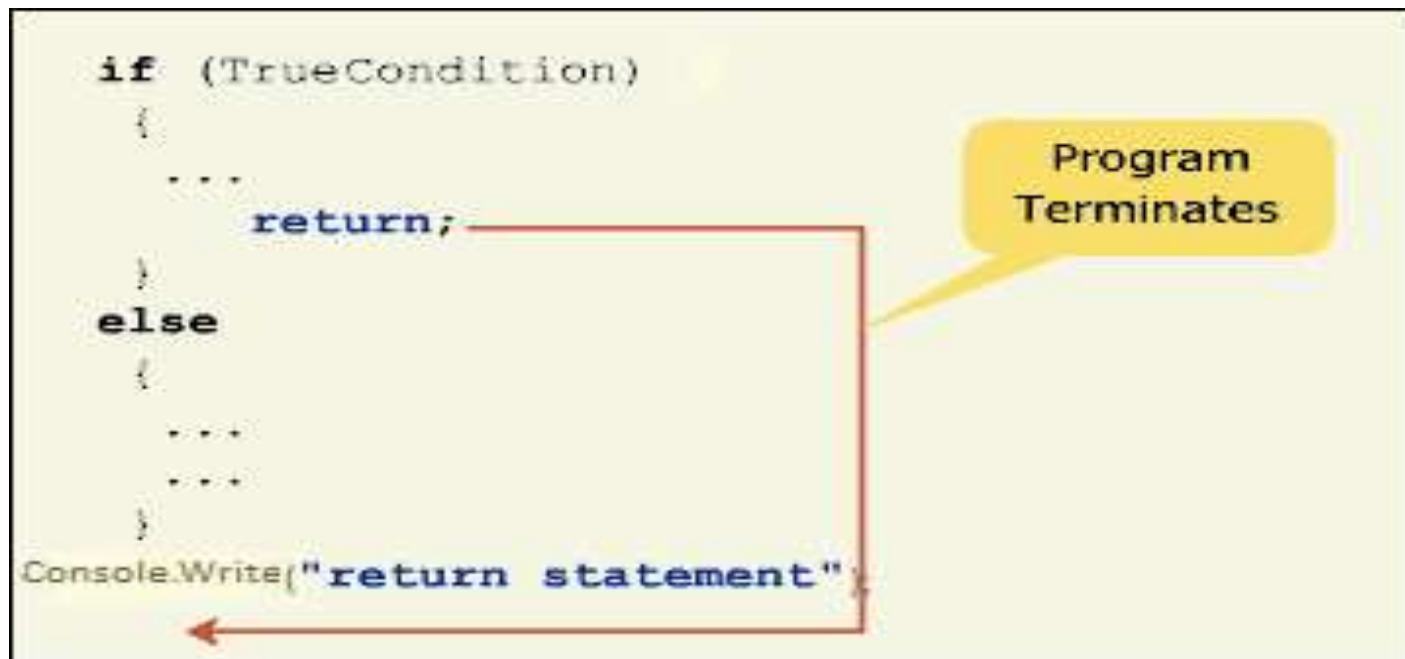
- ◆ allows to directly execute a labeled statement or block of statements.
- ◆ A labeled block or statement starts with a label.
A label is an identifier ending with a colon.
- ◆ A single labeled block can be referred by more than one **goto** statements.



Snippet

```
int i = 0;
display:
    Console.WriteLine("Hello World");
    i++;
    if (i < 5) {
        goto display;
    }
```

- ◆ used to return a value of an expression or is used to transfer the control to the invoking method.
- ◆ must be the last statement in the method block.



- ◆ Selection constructs are decision-making blocks that execute a group of statements based on the boolean value of a condition.
- ◆ C# supports if...else, if...else...if, nested if, and switch...case selection constructs.
- ◆ Loop constructs execute a block of statement repeatedly for a particular condition.
- ◆ C# supports while, do-while, for, and foreach loop constructs.
- ◆ The loop control variables are often created for loops such as the for loop.
- ◆ Jump statements transfer the control to any labeled statement or block within a program.
- ◆ C# supports break, continue, goto, and return jump statements.