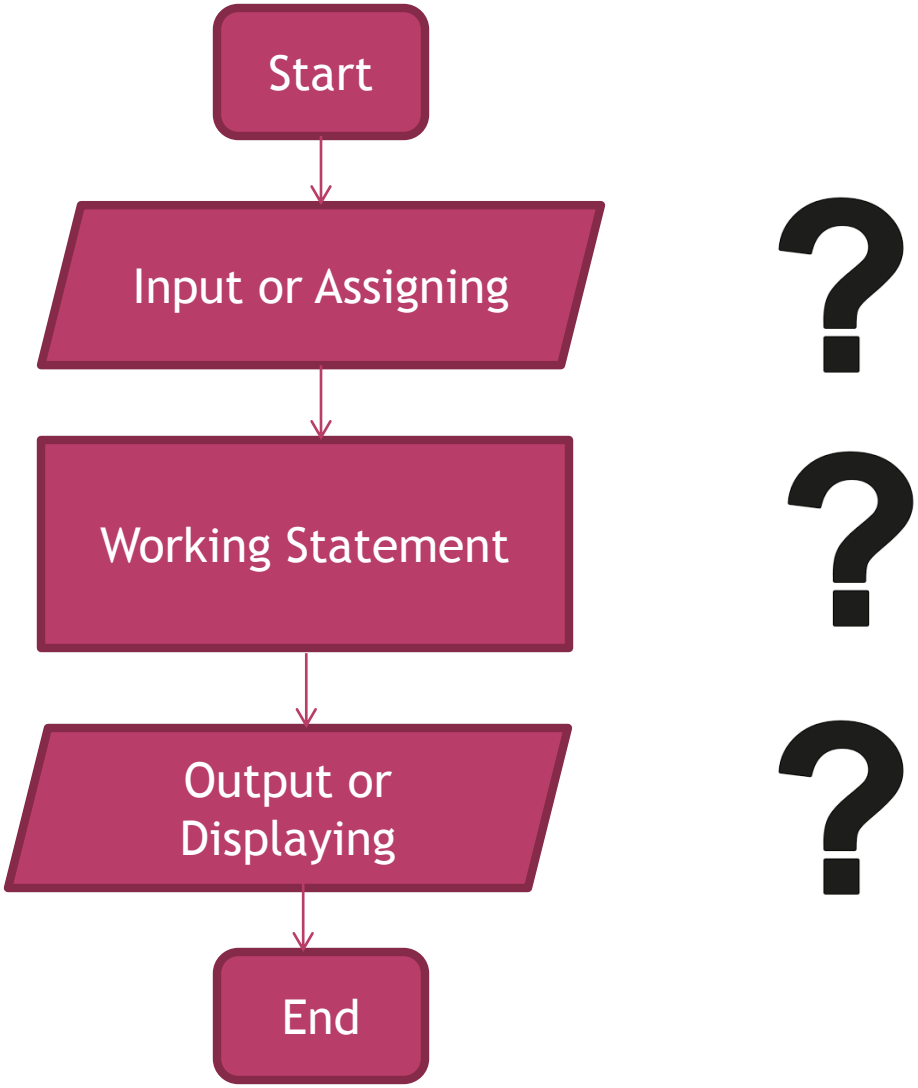
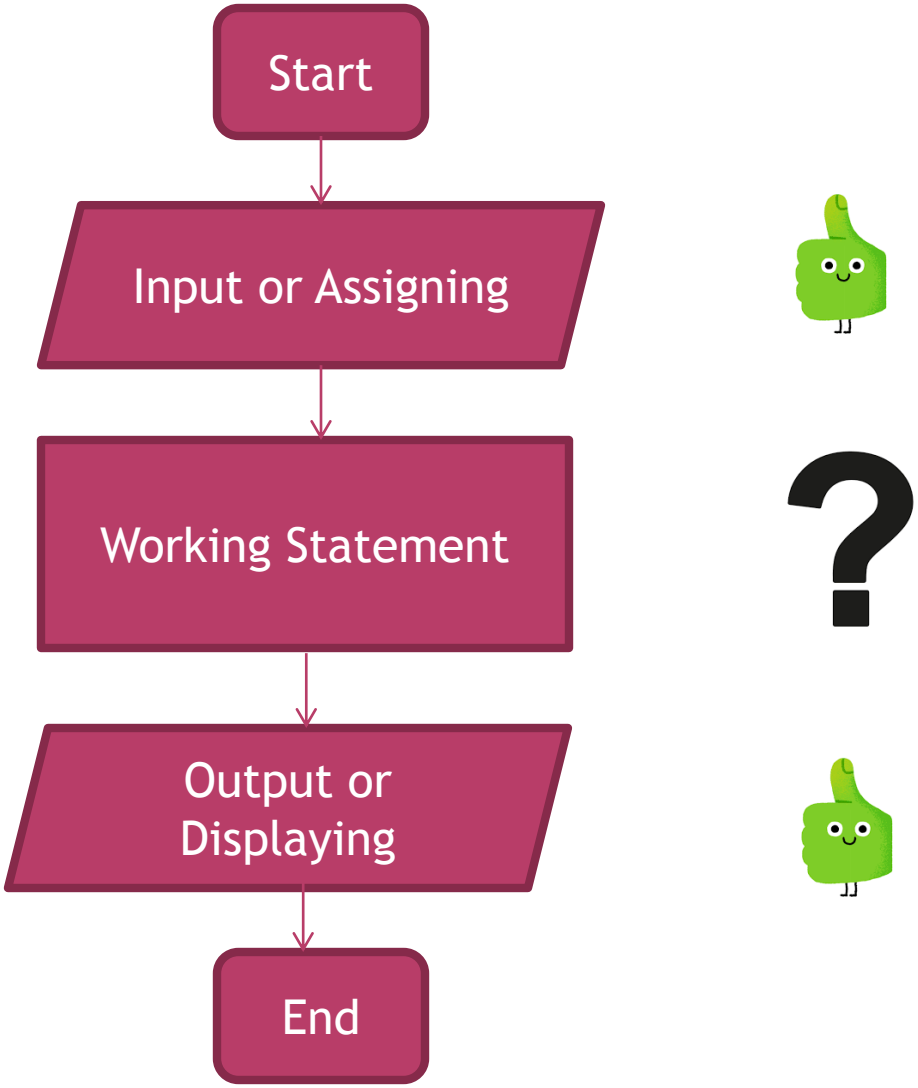
WEEK 3
LOOPS AND DECISIONS

Dr. Yuzana Win

COMPONENT OF A PROGRAM



COMPONENT OF A PROGRAM



RELATIONAL OPERATORS

- A relational operator compares two values.

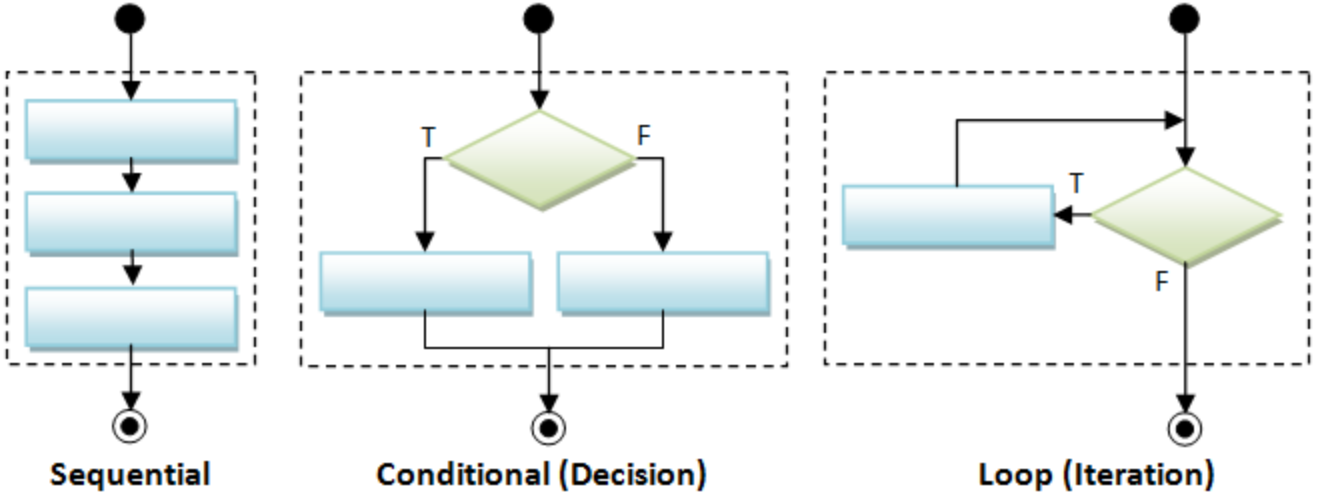
Operator	Description	Usage	Example (x=5, y=8)
==	Equal to	*expr1* == *expr2*	(x == y) → false
!=	Not Equal to	*expr1* != *expr2*	(x != y) → true
>	Greater than	*expr1* > *expr2*	(x > y) → false
>=	Greater than or equal to	*expr1* >= *expr2*	(x >= 5) → true
<	Less than	*expr1* < *expr2*	(y < 8) → false
<=	Less than or equal to	*expr1* >= *expr2*	(y <= 8) → true

RELATIONAL OPERATORS

```
// demonstrates relational operators
#include <iostream>
using namespace std;
int main()
{
        int numb;
        cout << “Enter a number:”;
        cin >> numb;
        cout << “numb<10  is ” << (numb < 10)  << endl;
        cout << “numb>10  is ” << (numb > 10)  << endl;
        cout << “numb==10 is ” << (numb == 10) << endl;
        return 0;
}
```

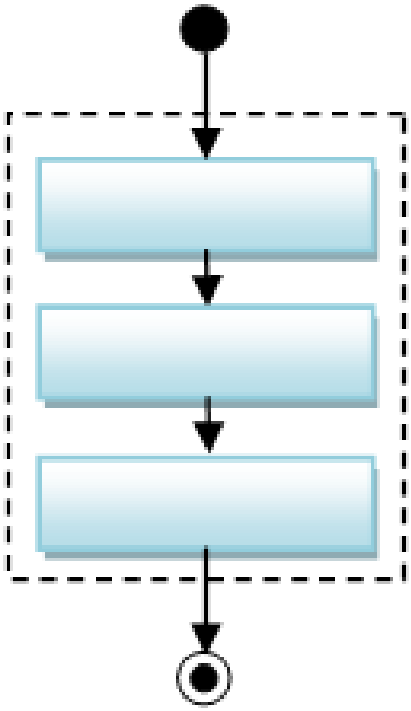
FLOW CONTROL

- **Three** basic flow control constructs:
  - (1) sequential
  - (2) conditional (or decision)
  - (3) loop (or iteration)



(1) SEQUENTIAL FLOW CONTROL

- A program is a sequence of instructions.
- Step-by-step



Sequential

(2) CONDITIONAL (DECISION) FLOW CONTROL

- Employs a number of conditions, which lead to a selection of one out of several alternative statement.
- Few types of conditionals,

  (1) if – then statement

  (2) if – then – else statement

  (3) nested – if (if – elseif – elseif – .. – else)

  (4) switch – case

  (5) conditional expression

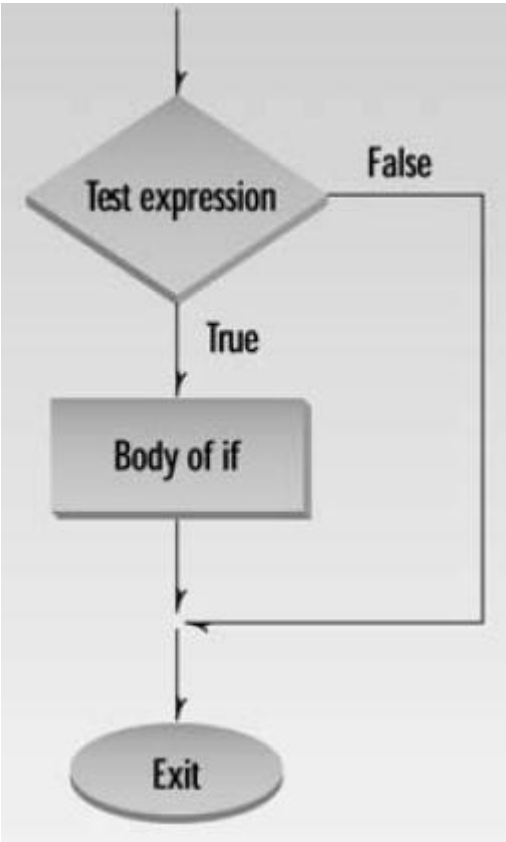
THE IF STATEMENT

- The if statement is the simplest of the decision-making statements.

Syntax

```
if <expression>
  statement;
-----------------------

if <expression>
{
    statement;
    statement;
    statement;
}
```



QUIZ

1. Calculate $y = x^2$ for $x < 5$
- Calculate $y = x^2$ for $x >= 5$ and $x < 10$
- Calculate $y = 3x + 7$ for $x > 5$

2. Write a program that can ask your name and tell you if your name is match or not match with the name that it has known. Note that the name it has known is “Hla Hla”.

EXAMPLE

```
// demonstrates if statement
#include <iostream>
using namespace std;
int main()
{
        int x;
        cout << “Enter a number: “;
        cin >> x;
        if( x > 100 )
        cout << “That number is greater than 100\n”;
        return 0;
}
```

MULTIPLE STATEMENTS

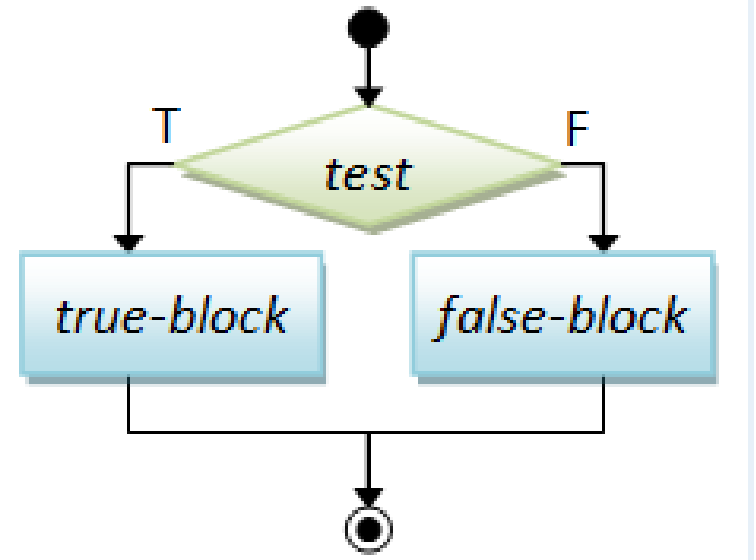
```
// if2.cpp
// demonstrates if with multiline body
#include <iostream>
using namespace std;
int main()
{
    int x;
    cout << “Enter a number: “;
    cin >> x;
    if( x > 100 )
    {
        cout << “The number” << x;
        cout << “ is greater than 100\n”;
    }
    return 0;
}
```

THE IF - THEN - ELSE STATEMENT

- If the condition is true (nonzero), the first statement is executed, else the second statement is executed

Syntax

```
if (expression)
        statement1;
else
        statement2;
```



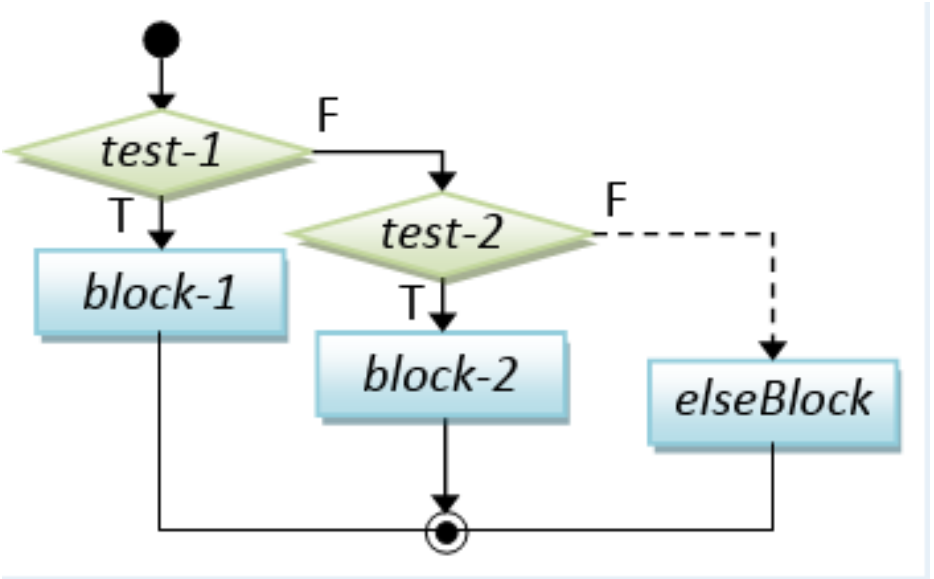
EXAMPLE

```
#include <iostream>
using namespace std;
int main()
{
        int x;
        cout << “Enter a number: “;
        cin >> x;
        if( x > 100 )
           cout << “That number is greater than 100\n”;
       else
           cout << “That number is not greater than 100 \n”;
       return 0;
}
```

THE NESTED - IF STATEMENT

Syntax

```
if  (expression)
    statement1;
else if (expression)
    statement2;
else
    statement3;
```



EXAMPLE

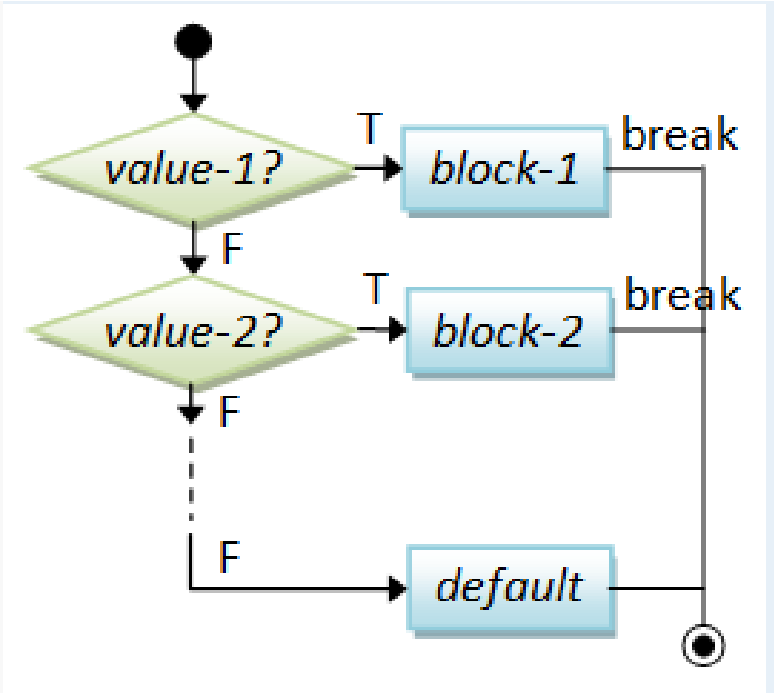
```
#include <iostream>
using namespace std;
int main()
{
        int mark;
        cout << “Enter a mark:”;
        cin >> mark;
       if (mark >= 80) {
           cout << "A" << endl;
       } else if (mark >= 70) {
           cout << "B" << endl;
       } else if (mark >= 60) {
           cout << "C" << endl;
       } else if (mark >= 50) {
           cout << "D" << endl;
       } else {
           cout << "F" << endl;
       }
       return 0; }
```

SWITCH - CASE STATEMENT

- **switch statement** is a conditional control statement that allows some particular group of statements to be chosen from several available groups.
- In a switch-case statement, a break statement is needed for each of the cases

Syntax

```
switch ( selector ) {
   case value-1:
      block-1; break;
   case value-2:
      block-2; break;
   case value-3:
      block-3; break;
   ......
   case value-n:
      block-n; break;
   default:
      default-block;
}
```



EXAMPLE

```
#include <iostream>
using namespace std;
int main()
{
        int  num;
        cout << " Please enter a number between 3 to 7:";
        cin >> num;
       switch (num)
       {
            case 3: cout << "The number is three"; break;
            case 4: cout << "The number is four"; break;
            case 5: cout << "The number is five"; break;
            case 6: cout << "The number is six"; break;
            case 7: cout << "The number is seven"; break;
            default: cout << "It is one of the undefined value";
         }
       return 0;
}
```

CONDITIONAL EXPRESSION

- A conditional operator is a ternary (3-operand) operator
- Ternary operator (?:) is more efficient form for expressing simple if statements
- It has the following form:

  expression1 ? expression2: expression3
- It simply states:

  if expression1 then expression2 else expression3
- For example to assign the maximum of a and b to c:

```
c = (a>b) ? a : b;
```

which is the same as:

```
if (a>b)
   c = a;
else
   c = b;
```

SUMMARY (SELECTION STRUCTURE)

- if statement is a condition based decision making statement.
- Ternary operator is more efficient form for expressing simple if statements
- The C++ switch statement is a conditional control statement that allows some particular group of statements to be chosen from several available groups.

QUIZ

1. Write a simple program that works like a ATM (Automated Teller Machine). The program should check the bank account number and the user's pin code (password) step by step. If two of them are corrected, the program should ask how much money you would like to withdraw.

(3) LOOPS (OR ITERATION)

- Looping statements allow the program to **repeat a certain number of times** or until some condition occurs.
- There are three types of loops

  (1) for loop ,

  (2) while loop,

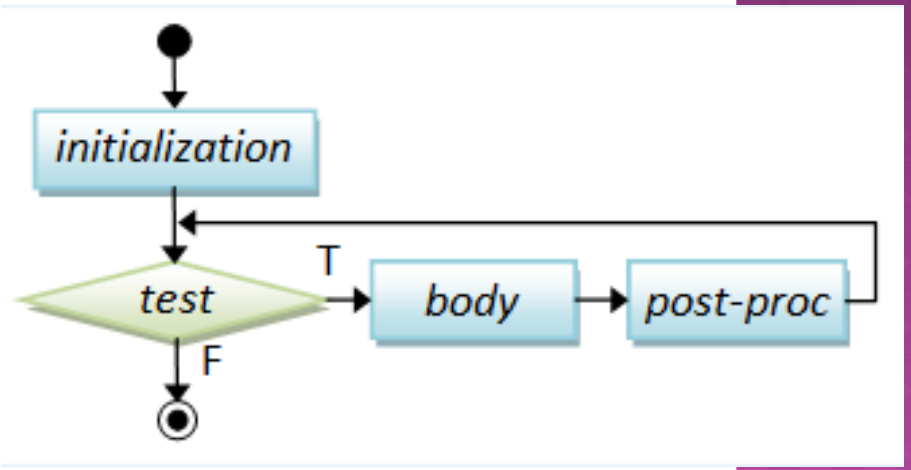
  (3) do-while loop

THE *FOR* LOOP

- The for loop executes a section of code a fixed number of times
- Usually used when you know, before entering the loop, how many times you want to execute the code

Syntax

```
// for-loop
for (init; test; post-proc) {
   body ;
}
```



EXAMPLE

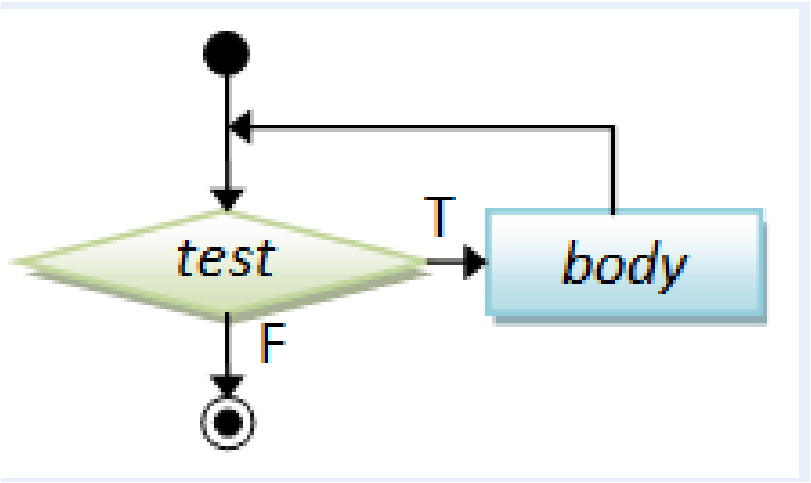
```
// demonstrates simple FOR loop
// displays the squares of the numbers from 0 to 14
#include <iostream>
using namespace std;
int main()
{
        int j;                          //define a loop variable
        for(j=0; j<15; j++)             //loop from 0 to 14,
        cout << j * j << " ";           //displaying the square of j
        cout << endl;
        return 0;
}
```

THE *WHILE* LOOP

- The while statement is used when the programs needs to perform repetitive tasks
- If you don't know how many times you want to do something, while loop is used

Syntax

```
// while-do
while ( condition ) {
   body ;
}
 <OR>
while (expression) {
    statement;
}
```



THE *WHILE* LOOP

- In a while loop, the test expression is evaluated *at the beginning* of the loop
- the **while** loops can accept expressions, not just conditions, the following are all legal:

```
while (x--);
while (x = x+1);
while (x += 5);
```

- Using this type of expression, only when the result of

x--, x=x+1, or x+=5,

evaluates to 0 will the **while** condition fail and the loop be exited.

EXAMPLE

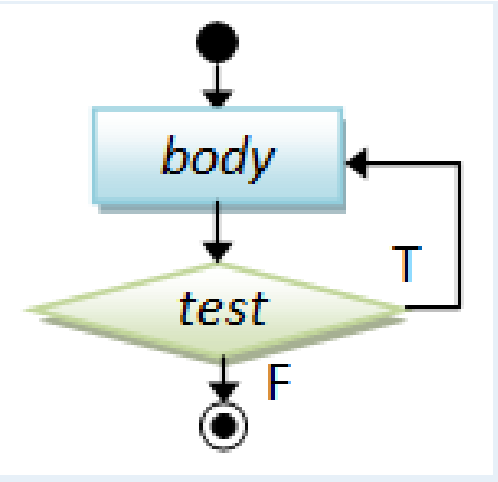
```
// demonstrates WHILE loop
#include <iostream>
using namespace std;
int main()
{
    int n = 99;      // make sure n isn't initialized to 0
    while( n != 0 )   // loop until n is 0
        cin >> n;       // read a number into n
    cout << endl;
    return 0;
}
```

THE *DO - WHILE* LOOP (REPEAT-UNTIL)

- Want to guarantee that the loop body is executed at least once, no matter what the initial state of the test expression
- Use the do loop
- The test expression is *at the end of the loop*

Syntax

```
// do-while
do{
    statement;
} while (expression);
 <OR>
do{
    <block of  statements>
} while (expression);
```



EXAMPLE

```
// divdo.cpp
// demonstrates DO loop
#include <iostream>
using namespace std;
int main()
{
long dividend, divisor;
char ch;
do                                  //start of do loop
{                                  //do some processing
   cout << “Enter dividend:”; cin >> dividend;
   cout << “Enter divisor:”;  cin >> divisor;
   cout << “Quotient is”<< dividend / divisor;
   cout << “, remainder is”<< dividend % divisor;
   cout << “\nDo another? (y/n):”;  //do it again?
   cin >> ch;
}
while( ch != ‘n’ );                    //loop condition
return 0; }
```

Output:

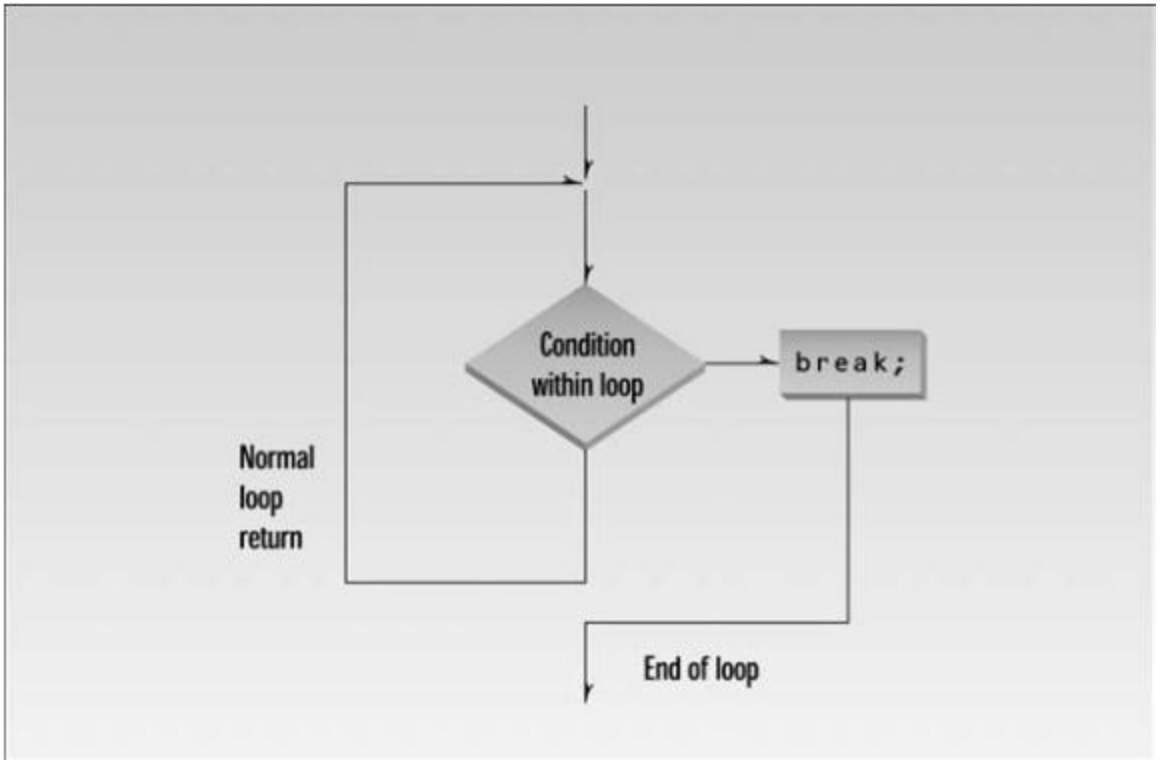
```
Enter dividend: 11
Enter divisor: 3
Quotient is 3,
Remainder is 2
Do another? (y/n): y
Enter dividend: 222
Enter divisor: 17
Quotient is 13,
remainder is 1
Do another? (y/n): n
```

OTHER CONTROL STATEMENT

- break statement
- continue statement

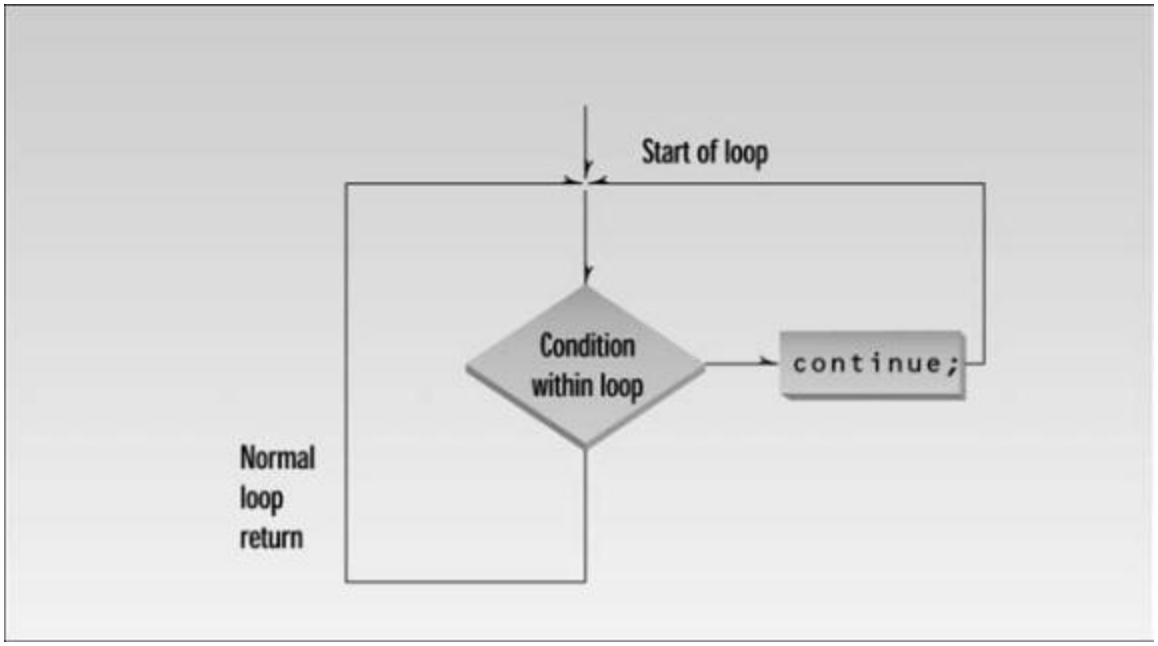
THE BREAK STATEMENT

- The break statement causes an exit from a loop



THE CONTINUE STATEMENT

- The continue statement aborts the current iteration and continue to the next iteration of the current (innermost) loop



EXAMPLE

```
// demonstrates CONTINUE statement
#include <iostream>
using namespace std;
int main()
{
long dividend, divisor;
char ch;
do {
        cout << "Enter dividend:"; cin >> dividend;
        cout << "Enter divisor:";  cin >> divisor;
        if( divisor == 0 )                          //if attempt to
            {                                       //divide by 0,
                cout << "Illegal divisor\n";
                continue;                       //go to top of loop
             }
        cout << "Quotient is"<< dividend / divisor;
        cout << ", remainder is"<< dividend % divisor;
        cout << "\nDo another? (y/n):";
        cin >> ch;
    } while( ch != 'n' );
return 0; }
```

SUMMARY (CONTROL STATEMENTS)

- Looping allows the program to repeat a section of code any number of times or until some condition occurs
- *for* statement is used to repeat a set of statements specified number of times
- *while* and *do..while* statements are repetitive control structures, that are used to carry out conditional looping

QUIZ

- Write a program that guess the number and type of roots of a quadratic equation ($ax^2+bx+c=0$) according to its discriminant (b^2-4ac)

  Discriminant

$b^2-4ac > 0$	two real numbers
$b^2-4ac = 0$	one real number
$b^2-4ac < 0$	two complex numbers

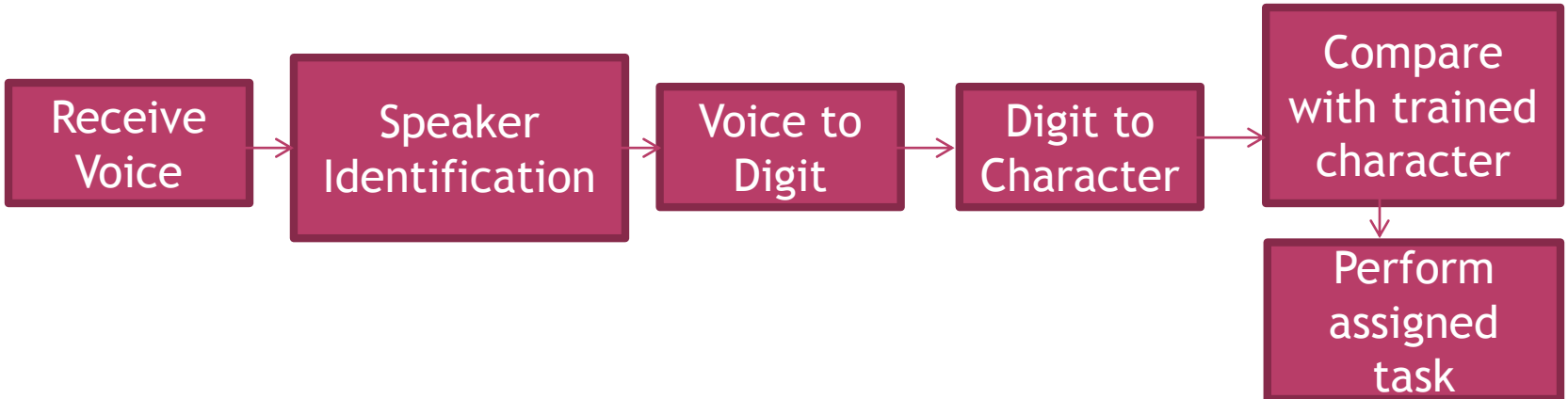
QUIZ

- Using "Switch Statement", write a program that works like a voice recognizer for some specific tasks trained as follows. The digit will be input by user in this example.

  Open door – 01111001

  Start engine – 1111110011

  Play music - 111011011



QUIZ

- Using “Switch Statement”

Cocacola	1
Max	2
Ve Ve	3
Red Bull	4
Sprite	5
7 Up	6
Asia	7

QUIZ

- Using “Switch Statement”

September 2017

Calendarpedia
Your source for calendars

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

Mon	4, 11, 18, 25
Tue	5, 12, 19, 26
Wed	6, 13, 20, 27
Thur	7, 14, 21, 28
Fri	1, 8, 15, 22, 28
SAT	2, 9, 16, 23, 30
SUN	3, 10, 17, 24

QUIZ

- Write a program to calculate the total sum of given first number and last number.

Total = first + (first+1) + (first+2)+ …+(last-2)+(last-1) + last

e.g.,

first number = 1

last number = 5

Total = 1+2+3+4+5=15

QUIZ

- Write a program that works like a cash register. Use a sample of products given in the following table. Product code must be input by the user in this example. By using

  (i) switch statement

  (ii) for- loop statement

  (iii) setw statement

Cash Register

Product Code	Item Name	Price
40200	Book	500
81007	Pen	350
90660	Pencil	150
53111	Correction Pen	650

QUIZ

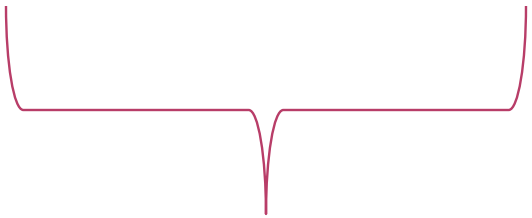
- Write a program to calculate how long it will take to attain the total balance more than 50000 started from a saving amount of 10000 with an interest rate of 10% per year as follows. (Using while loop)

Total Balance = Save * (1+0.1)

QUIZ

- Write a program to calculate addition of 2 for twenty times (using break command)

  Total = 2 + 2+ 2 + ---- + 2



20

QUIZ

- Write a program that suddenly break the loop of escalator when it is overloaded or the user switch off. (using break command)
  - Maximum load is random
  - The allowable maximum load = 3 tons

QUIZ

- Consider the example where we read an integer values and process them according to the following conditions.

- If the value we have read is negative, we wish to print an error message and abandon the loop.

- If the value read is greater than 100, we wish to ignore it and continue to the next value in the data.

- If the value is zero, we wish to terminate the loop.

ASSIGNMENTS

1. Write a C++ program that takes number of rows and number of columns as input and print stars according to number of rows and columns.
2. Write a C++ program that takes number of value to get as input and find maximum value among input values.
3. Write a C++ program that takes size of multiplication table and prints multiplication table with that size starting from 2.
4. Write a program that display the prime numbers starting from 2 to 100.