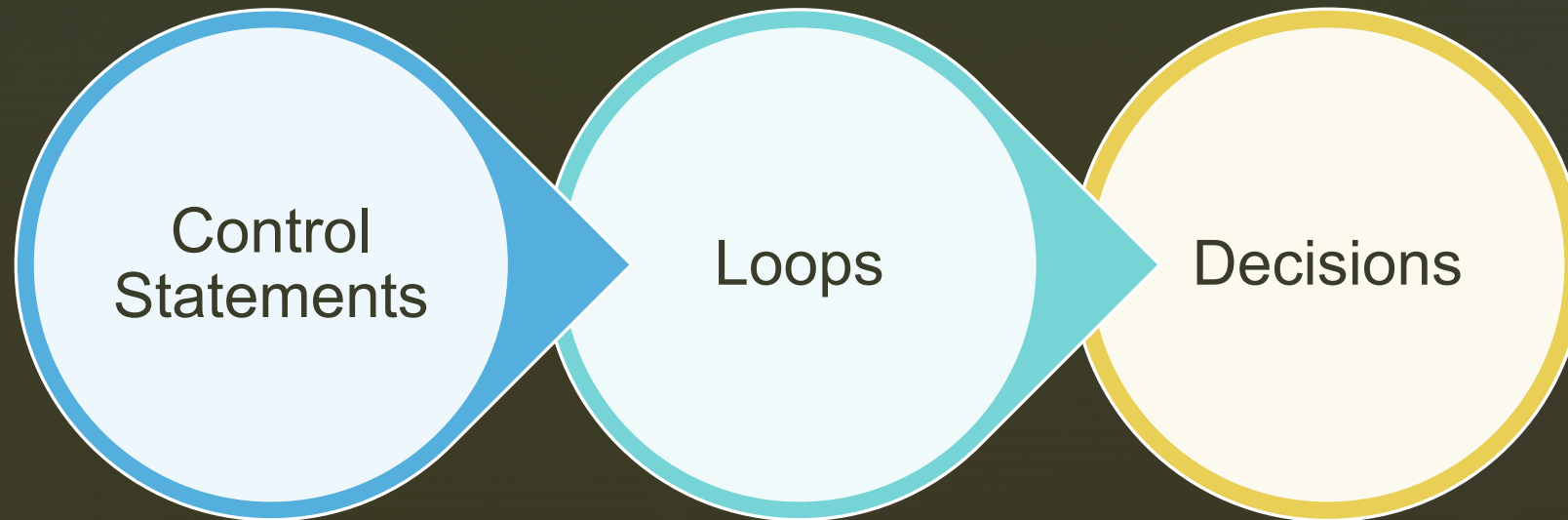


DR. (MRS) T-S.M.A. ADJAIDOO

02

Flow of Control

Today's Lesson



Learning Outcomes

1

- Understanding what flow of control is

2

- How to make decisions in C++

3

- How to make iterations in C++

Flow of Control



[This Photo](#) by Unknown Author is licensed under [CC BY-SA](#)



Control Statements

- Not many programs execute all their statements in strict order from beginning to end.
- Most programs (like many humans) decide what to do in response to changing circumstances.
- The flow of control jumps from one part of the program to another, depending on calculations performed in the program.
- Program statements that cause such jumps are called control statements.
- There are two major categories: loops and decisions.

Control Statements

- How many times a loop is executed, or whether a decision results in the execution of a section of code, depends on whether certain expressions are true or false.
- These expressions typically involve a kind of operator called a *relational operator*, which compares two values

Relational Operators

Operator	Meaning
>	Greater than
<	Less than
==	Equal to
!=	Not equal to
>=	Greater than or equal to
<=	Less than or equal to

Loops

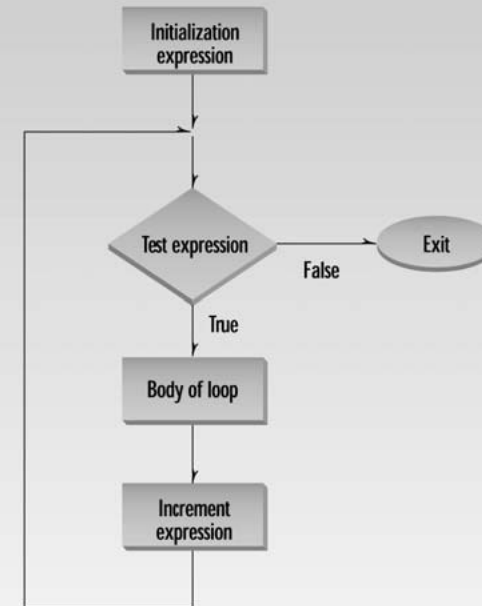
- There are three kinds of loops in C++:
 - the for loop
 - the while loop
 - the do-while loop

For Loop

a) `for (j=0; j<15; j++)` — Note: no semicolon here
`statement;` — Single-statement loop body

b) `for (j=0; j<15; j++)` — Note: no semicolon here
`{`
`statement;`
`statement;`
`statement;`
`}` — Multiple-statement loop body—
a block of code
— Note: no semicolon here

Initialization expression
Test expression
Increment expression



For Loop

```
// demonstrates simple FOR loop
#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;
}
```

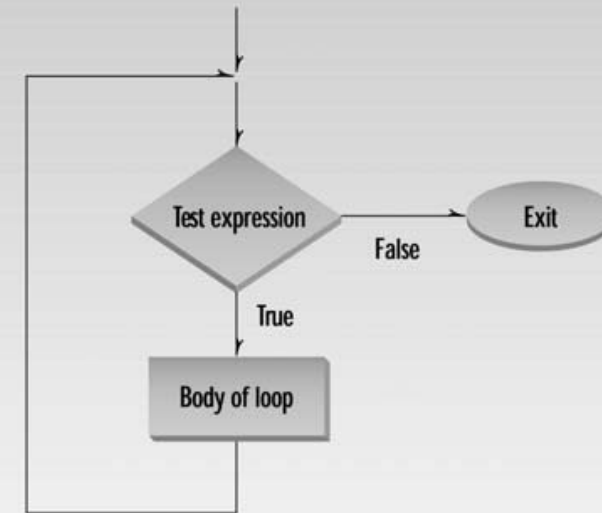
While Loop

- The for loop does something a fixed number of times.
- What happens if you don't know how many times you want to do something before you start the loop?
- In this case a different kind of loop may be used:
 - the while loop.

While Loop

Test expression
`while (n!=0) {` — Note: no semicolon here
 `statement;` — Single-statement loop body
`}`

Test expression
`while (v2<45) {` — Note: no semicolon here
 `statement;`
 `statement;` } Multiple-statement loop body
 `statement;`
 `}` — Note: no semicolon here



While Loop

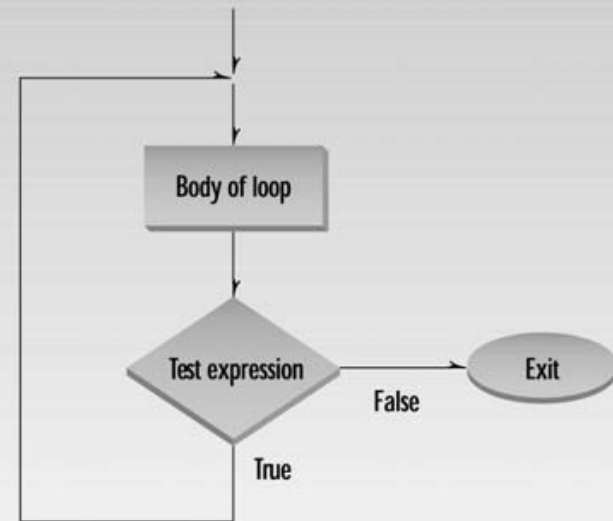
```
// demonstrates WHILE loop
#include <iostream>
using namespace std;

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;
    }
}
```

Do-while loop

`do` — Note: no semicolon here
 `statement;` — Single-statement loop body
`while (ch != 'n');`
 Test expression — Note: semicolon

`do` — Note: no semicolon here
 {
 `statement;`
 `statement;`
 `statement;` } — Multiple-statement loop body
 }
`while (numb < 96);`
 Test expression — Note: semicolon



Do-while loop

```
// 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;
}
```

Loops

- The **for loop** is appropriate when you know in advance how many times the loop will be executed.
- The **while** and **do-while loops** are used when you don't know in advance when the loop will terminate.

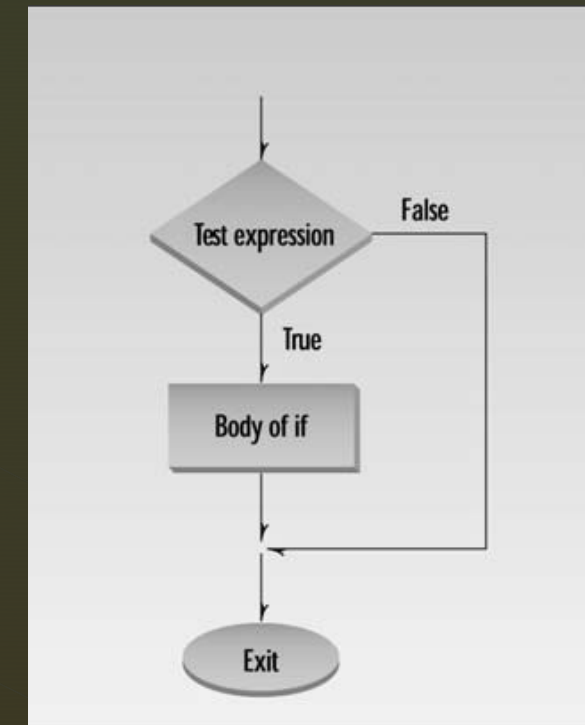
Decisions

- Decisions can be made in C++ in several ways.
- The most important is with the `if...else` statement, which chooses between two alternatives.
- This statement can be used without the `else`, as a simple `if` statement.
- Another decision statement, `switch`, creates branches for multiple alternative sections of code, depending on the value of a single variable.
- Finally, the `conditional operator` is used in specialized situations.

If statement

Test expression
if (x > 100)
statement; Single-statement if body

Test expression
if (speed <= 55)
{
statement;
statement;
statement;
}
Note: no semicolon here
Multiple-statement if body



If statement

```
// 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;
}
```

If ... else statement

Test expression

```
if (x>100)
    statement;
else
    statement;
```

Single-statement if body

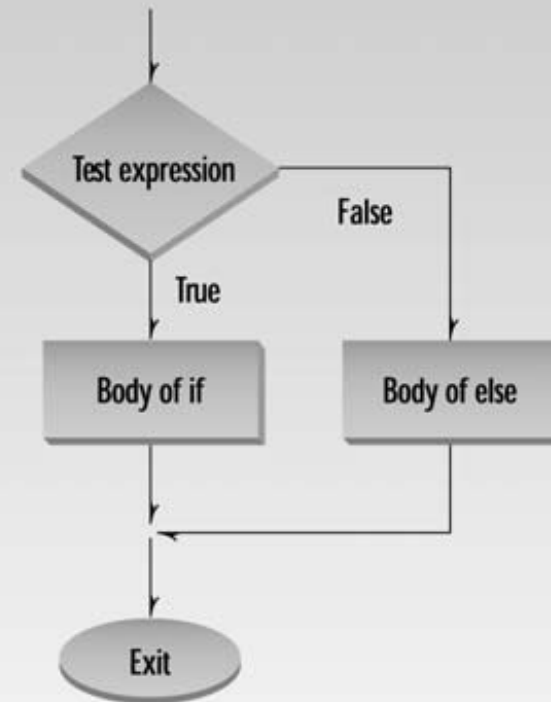
Single-statement else body

Test expression

```
if (zebra!=0)
{
    statement;
    statement;
}
else
{
    statement;
    statement;
}
```

Multiple-statement if body

Multiple-statement else body



If...else statement

```
// demonstrates IF ... ELSE 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";
    else
        cout << "That number is less than 100\n";
    return 0;
}
```

Switch Statement

```
switch (n) {  
    case 1:  
        statement;  
        statement;  
        break;  
    case 2:  
        statement;  
        statement;  
        break;  
    case 3:  
        statement;  
        statement;  
        break;  
    default:  
        statement;  
        statement;  
}
```

Integer or character variable

Note: no semicolon here

Integer or character constant

First case body

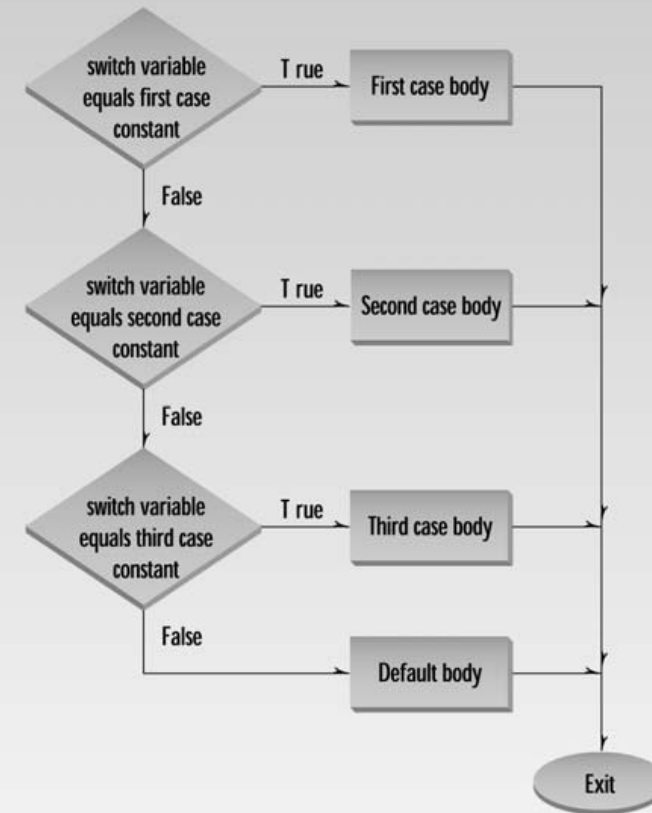
causes exit from switch

Second case body

Third case body

Default body

Note: no semicolon here



Switch Statement

```
// demonstrates SWITCH statement
#include <iostream>
using namespace std;

int main()
{
    int speed; //turntable speed
    cout << "\nEnter 33, 45, or 78: ";
    cin >> speed; //user enters speed
    switch(speed) //selection based on speed
    {
        case 33: //user entered 33
            cout << "LP album\n";
            break;

        case 45: //user entered 45
            cout << "Single selection\n";
            break;

        case 78: //user entered 78
            cout << "Obsolete format\n";
            break;
    }
    return 0;
}
```

Switch Statement

Let's look at the switch
adventure Program

```
// demonstrates SWITCH with adventure program
#include <iostream>
using namespace std;
// #include <conio.h> //for getche()

int main()
{
    char dir = 'a';
    int x=10, y=10;
    while( dir != '\r' )
    {
        cout << "\nYour location is " << x << ", " << y;
        cout << "\nEnter direction (n, s, e, w): ";
        dir = getchar(); //get character

        switch(dir) //switch on it
        {
            case 'n': y--; break; //go north
            case 's': y++; break; //go south
            case 'e': x++; break; //go east
            case 'w': x--; break; //go west
            case '\r': cout << "Exiting\n";
                       break; //Enter key
            default: cout << "Try again\n"; //unknown char
        } //end switch
    } //end while
    return 0;
} //end main
```


Conditional Operator

Conditional expression

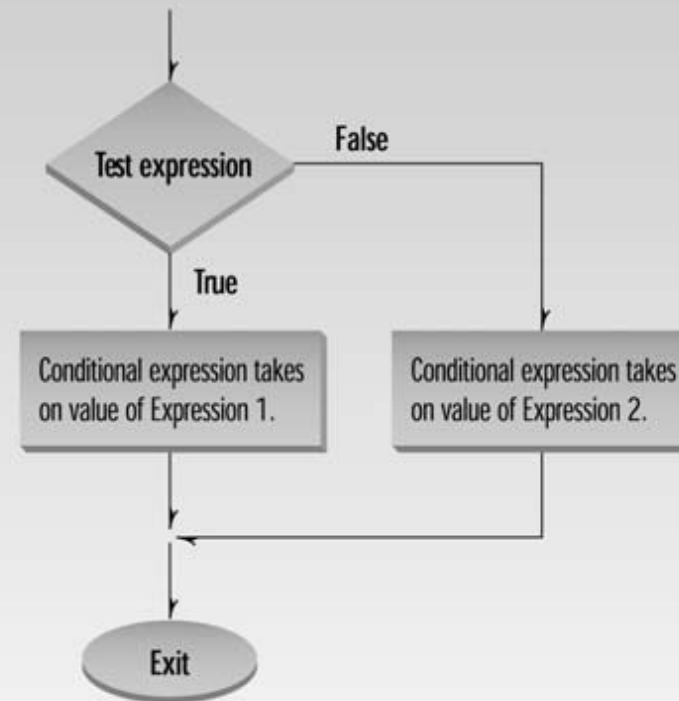
```
result = (alpha < 77) ? beta : gamma;
```

Test expression

Expression 1

Expression 2

Conditional operator



Logical Operators

Operator	Effect
&&	Logical AND
	Logical OR
!	Logical NOT

Exercises

- Write a temperature-conversion program that gives the user the option of converting Fahrenheit to Celsius or Celsius to Fahrenheit. Then carry out the conversion.
- Create the equivalent of a four-function calculator.
 - The program should ask the user to enter a number, an operator, and another number.
 - It should then carry out the specified arithmetical operation: adding, subtracting, multiplying, or dividing the two numbers.
 - Use a switch statement to select the operation.
 - Finally, display the result.
 - When it finishes the calculation, the program should ask whether the user wants to do another calculation. The response can be 'y' or 'n'.



Any Questions?

The End

Contact: tsadjaidoo@knust.edu.gh

Office: Caesar Building, Room 413