Objectives

In this chapter, you will learn about:

- Conditions
- The if-else statement
- The switch statement

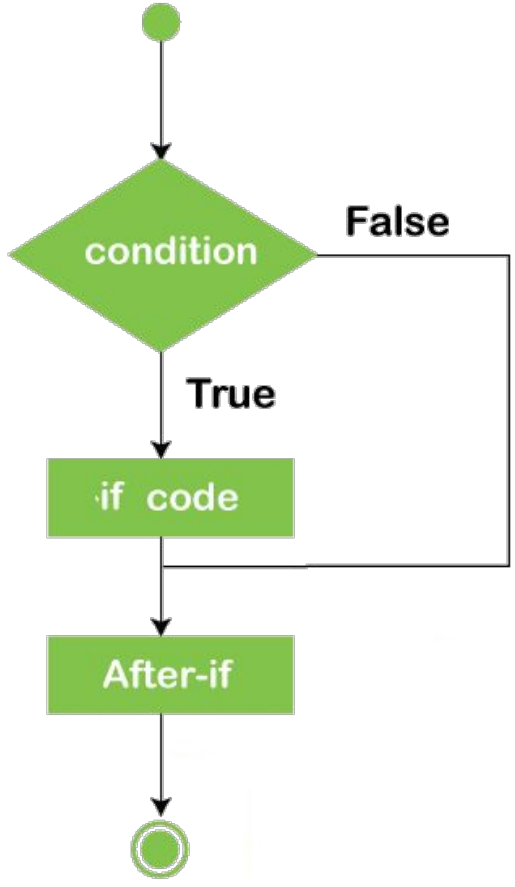
The if... Statement

The if statement is the most basic conditional structure, and it can be found in all programming languages (with different syntax). It allows executing a series of instructions if a condition is true.

Example:

```
if  ( condition )
{
      //  List  of  instructions ;
}

if  ( grade  >=  10)
{
      printf ("You  are  admitted !");
}
```



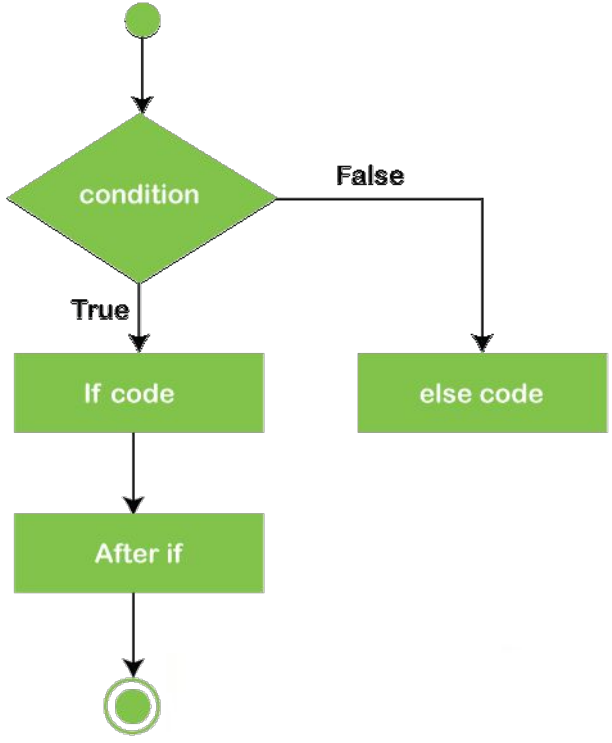
The if...else... Statement

The if...else statement allows executing another series of instructions if the condition is not true.

Example:

```
if  ( condition )
  {
  //  List  of  instructions ;
  }

else
  {
  //  Another  block  of  instructions ;
  }
```



Conditions

Conditions refers to an expression or a combination of expressions built using comparison operators.

Symbole	Signification
==	est égal à
>	est supérieur à
<	est inférieur à
>=	est supérieur ou égal à
<=	est inférieur ou égal à
!=	est différent de

Example:

```
int age = 20;

if ( age >= 18) {
      // This is a condition : ( age >= 18)
      printf (" You are an adult .\n");
} else {
      printf (" You are a minor .\n");
}
```

Logical Operators

Logical operators allow combining logical expressions (conditions).

Symbole	Signification
&&	ET
||	OU
!	NON

Example:

```
int main () {
        int x = 5;
        int y = 10;
        int z = 5;
        int i1 = 1;
        int i2 = 0;

        if (x == y && x == z) {
                printf ("x is equal to y and z.\n");
        } else if (x == y || x == z) {
                printf ("x is equal to either y or z (or both ).\n");
        } else if (!( x == y)) {
                printf ("x is not equal to y.\n");
        }

                printf ("i1 || i2 = %d", i1 || i2);
                printf ("i1 && i2 = %d", i1 && i2);
                printf (" negation (1) = %d", !(1) );
}
```

Nested conditional statements 1/3

We want a program to determine whether the integer given by the user is positive, negative, or zero and displays the corresponding message

Example:

```
int main () {
        int value ;

        printf (" Enter an integer : ");
        scanf ("%d", & value );

        if ( value > 0) {
                printf (" The value is positive .\n");
        }
        if ( value < 0) {
                printf (" The value is negative .\n");
        }
        if ( value == 0) {
                printf (" The value is zero .\n");
        }

        return 0;
}
```

Nested conditional statements 2/3

- A nested conditional statement refers to the practice of placing one if-else statement inside another if-else statement.
- This is done to create more complex and structured decision-making processes based on multiple conditions.

Example:

```
int main () {
    int value ;

    printf (" Enter an integer : ");
    scanf ("%d", & value );

    if ( value > 0) {
        printf (" The value is positive .\n");
    } else if ( value < 0) {
        printf (" The value is negative .\n");
    } else {
        printf (" The value is zero .\n");
    }

    return 0;
}
```

Nested conditional statements 3/3

- A nested conditional statement refers to the practice of placing one if-else statement inside another if-else statement.

processes based on multiple conditions.

Example:

```
int main () {
    int value ;

    printf (" Enter an integer : ");
    scanf ("%d", & value );

    if ( value > 0) {
        printf (" The value is positive .\n");
    } else {
        if ( value < 0) {
            printf (" The value is negative .\n");
        } else {
            printf (" The value is zero .\n");
        }
    }

    return 0;
}
```

The switch Statement (1/3)

The switch statement allows multiple tests on the values of the same variable.

Syntax:

```
switch ( Variable ) {
        case Value1 :
                // List of instructions ;
                break ;
        case Value2 :
                // List of instructions ;
                break ;
        case Values ... :
                // List of instructions ;
                break ;
        default :
                // List of instructions ;
}
```

The switch Statement (1/3)

```
int main () {
    int choice ;
    printf (" Menu :\n");
    printf ("1. New Game \n");
    printf ("2. Load Game \n");
    printf ("3. Options \n");
    printf ("4. Quit \n");
    printf (" Enter your choice : ");
    scanf ("%d", & choice );
    switch ( choice ) {
        case 1:
            printf (" Starting a new game ...\ n");
            break ;
        case 2:
            printf (" Loading a saved game ...\ n");
            break ;
        case 3:
            printf (" Opening options menu ...\ n");
            break ;
        case 4:
            printf (" Quitting the application ...\ n");
            break ;
        default :
            printf (" Invalid choice . Please select a valid option .\n");
            break ;
    }
}
```

A Shorter Way to Test

You can use a more concise structure to perform tests.

(condition) ? instruction if true : instruction if false

- The condition should be enclosed in parentheses.
- When the condition is true, the left instruction is executed.
- When the condition is false, the right instruction is executed.
- Additionally, the ?: structure returns the value resulting from the test.

Example:

```
if  ( average  >=  10)
        printf (" Admitted ");
else
                printf (" Failed ");


( average  >=  10) ?  printf (" Admitted ")  :  printf (" Failed ");


admitted  =  ( average  >=  10)  ?  1  :  0;
```

In Summary

- The value **True** can be equated to the numerical value **1** or any **non-zero** value.

- Don't forget the **parentheses** when using **if** statements.

Example 1:

```
if   (1)
{
        printf (" This   is   true ");
}
else
{
        printf (" This   is   false ");
}
```

An example to conclude

```
int main () {
    int hour = 0;

    // Prompt the user for input
    printf (" Enter the hour (0 -23) : ");
    scanf ("%d", & hour );

    if ( hour < 0 || hour > 23) {
        printf (" Invalid hour . Please enter a valid hour between 0
            and 23.\ n");
    } else {
        if ( hour >= 6 && hour < 12) {
            printf (" Good morning !\n");
        } else if ( hour >= 12 && hour < 17) {
            printf (" Good afternoon !\n");
        } else if ( hour >= 17 && hour < 20) {
            printf (" Good evening !\n");
        } else {
            printf (" Good night !\n");
        }
    }

    return 0;
}
```