

If Statements :

- The C programming language provides a general decision-making capability in the form of an “if statement”.

```
if (expression)
```

```
Program statement
```

- Translating a statement such as “If it is not raining , then I will go swimming” into the C language is easy

```
if (it is not raining)
```

```
    I will go swimming
```

- The if statement is used to stipulate execution of a program statement/s based upon specific conditions.
 - I will go swimming if it is not raining.
- The curly brackets are required for compound statements inside the if block.

If statement (example)

```
int score = 95;
```

```
int big = 90;
```

```
// simple statement if, no brackets
```

```
if (score > big)
    printf("Jackpot!\n");
```

```
// compound statement if, brackets
```

```
if (score > big)
{
    score++;
    printf("You win\n");
}
```

If with an else

- You can extend the if statement with a small addition that gives you a lot more flexibility.

```
if the rain today is worse than the rain yesterday,
```

```
I will take my umbrella.
```

```
Else
```

```
I will take my jacket.
```

```
Then I will go to work.
```

- This is exactly the kind of decision making the if-else statement provides.

```
if (expression)
    statement1;
```

```
else
    statement2;
```

If with an else (example)

- // Program to determine if a number is even or odd

```
#include <stdio.h>

int main ()
{
    int number_to_test, remainder;

    printf ("Enter your number to be tested: ");
    scanf ("%i", &number_to_test);

    remainder = number_to_test % 2;

    if ( remainder == 0 )
        printf ("The number is even.\n");
    else
        printf ("The number is odd.\n");

    return 0;
}
```

else if

- You can handle additional complex decision making by adding an if statement to else clause:

```
if (expression 1)
    program statement 1
else
    if (expressions 2 )
        program statement 2
else
    program statement 3
```

- The above extends the if statement from a two-valued logic decision to a three-values logic decision.
→ formatted using the else if construct.

else if example

```
// Program to implement the sign function
```

```
#include <stdio.h>
```

```
int main (void)
```

```
{
```

```
    int number, sign;
```

```
    printf ("Please type in a number: ");
```

```
    scanf ("%i", &number);
```

```
    if ( number < 0 )
```

```
        sign = -1;
```

```
    else if ( number == 0 )
```

```
        sign = 0;
```

```
    else // Must be positive
```

```
        sign = 1;
```

```
    printf ("Sign = %i\n", sign);
```

```
    return 0;
```

```
}
```

i

Nested if-else statement

- A nested if-else statement means you can use if or else if statement inside another if or else if statements(s)

```
if (boolean_expression 1)
```

```
{
```

```
// Executes when the boolean expression 1 is true //
```

```
if (boolean_expression2)
```

```
    { // Executes when the boolean expression2 is true//
```

```
    }
```

```
}
```

Nested if statement code example

```
if ( gamelsOver == 0 )
```

```
    if ( playerToMove == YOU )
```

```
        printf ("Your Move\n");
```

```
    else
```

```
        printf ("My Move\n");
```

```
else
```

```
    printf ("The game is over\n");
```

The Conditional operator. (ternary statement)

- The conditional operator is a unique operator
 - Unlike all other operators in C
 - Most operators are either unary or binary operators
 - Is a ternary operator (takes three operands)
- The two symbols that are used to denote this operator are the question(?) and the colon (:)
- The first operand is placed before the ?, the second between the :, and the third after the :
 - Condition ? expression1: expression2
- The conditional operator evaluates to one of two expressions, depending on whether a logical expression evaluates true or false.
- Notice how the operator is arranged in relation to the operands
 - The? Character follows the logical expression, condition
 - On the right of ? are two operands , expression1 and expression2, that represent choices.
 - The value that results from the operation will be the value of expression1 if condition evaluates to true , or the value of expression2 if condition evaluates to false.

`X = y > 7 ? 25 : 50;`

- Results in x being set to 25 if y is greater than 7 , or to 50 otherwise
- Same as :
If (y > 7)
 X = 25;
else
 X = 50;