

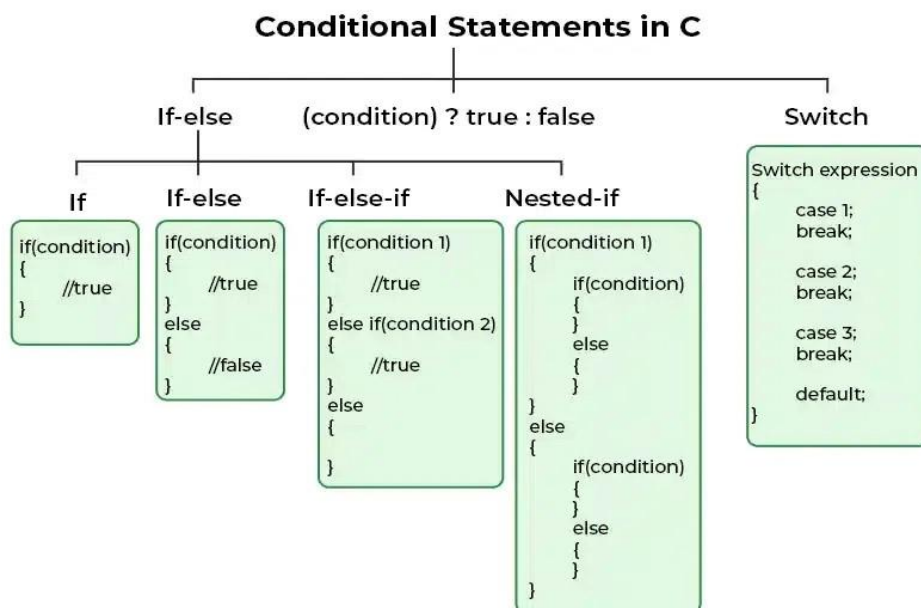
UNIT -2

Decision making and branching statements- Looping statement, Functions, call by value, call by reference.

Decision making statement in C

- The **conditional statements** (also known as decision control structures) such as if, if else, switch, etc. **are used for decision-making purposes** in C programs.
- They are also known as Decision-Making Statements and **are used to evaluate one or more conditions and make the decision whether to execute a set of statements or not.**
- *These decision-making statements in programming languages **decide the direction of the flow of program execution.**

Types of Conditional Statements in C



1. **if Statement**
2. **if-else Statement**
3. **Nested if Statement**
4. **if-else-if Ladder**
5. **switch Statement**
6. **Conditional Operator**
7. **Jump Statements:**

- **break**
- **continue**
- **goto**
- **return**

If in C

Syntax of if Statement

```
if(condition)
{
    // Statements to execute if
    // condition is true
}
```

- The **condition** after evaluation will be either true or false
- C if statement accepts boolean values – if the value is true then it will execute the block of statements below it otherwise not.
- As the condition present in the if statement is false. So, the block below the if statement is not executed.

Example program

```
#include <stdio.h>
int main()
{
    int i = 10;

    if (i > 15)
    {
        printf("10 is greater than 15");
    }

    printf("I am Not in if");
}
```

If-else

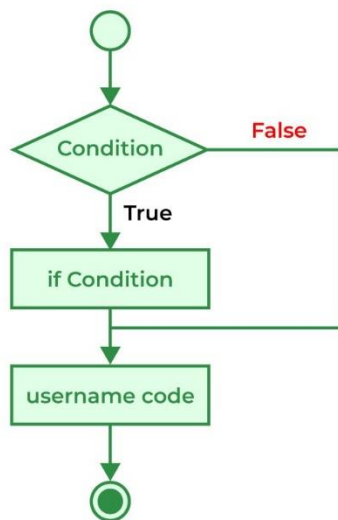
- The *else* statement with the *if* statement to execute a block of code when the condition is false.
- The [if-else statement](#) consists of two blocks, one for false expression and one for true expression.

Syntax of if else in C

```
if (condition)
{
    // Executes this block if
    // condition is true
}
else
{
    // Executes this block if
    // condition is false
}
```

Flowchart of if-else Statement

Flowchart of if Statement



Example program

```
#include <stdio.h>
int main()
{
    int i = 20;
    if (i < 15)
    {
        printf("i is smaller than 15");
    }
    else
    {
        printf("i is greater than 15");
    }
    return 0;
}
```

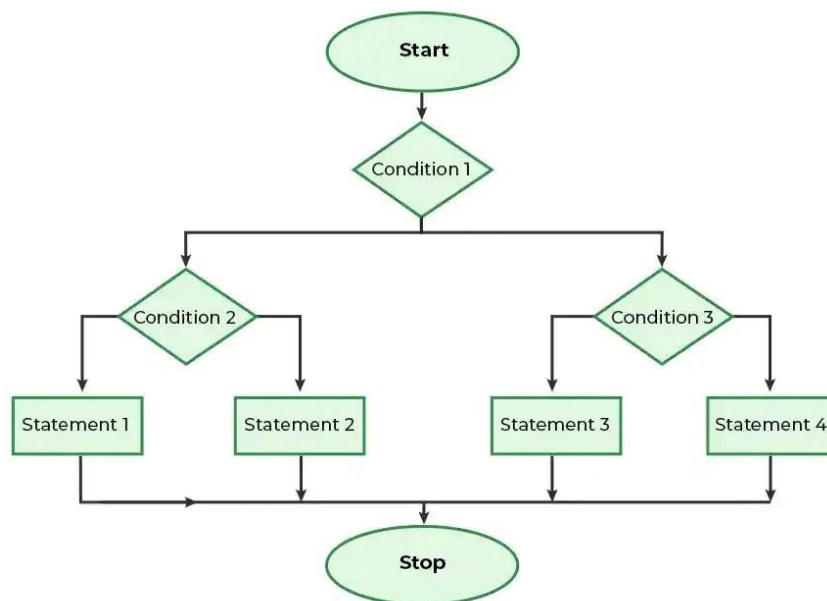
3. Nested if-else in C

- A nested if in C is an if statement that is the target of another if statement.
 - Nested if statements mean an if statement inside another if statement.
- Syntax of Nested if-else

Syntax

```
if (condition1)
{
    // Executes when condition1 is true
    if (condition2)
    {
        // Executes when condition2 is true
    }
    else
    {
        // Executes when condition2 is false
    }
}
```

Flowchart of Nested if-else



```
#include <stdio.h>
int main()
{
    int i = 10;
    if (i == 10) {
        // First if statement
        if (i < 15)
        {
            printf("i is smaller than 15\n");
        }
    }
}
```

```

// Nested - if statement
// Will only be executed if statement above
// is true
if (i < 12)
{
    printf("i is smaller than 12 too\n");
}
else
printf("i is greater than 15");
}

return 0;
}

```

4. if-else-if Ladder in C

- The [if else if statements](#) are used when the user has to decide among multiple options.
- The C if statements are executed from the top down. As soon as one of the conditions controlling the if is true, the statement associated with that if is executed, and the rest of the C else-if ladder is bypassed.
- If none of the conditions is true, then the final else statement will be executed.
- If-else-if ladder is similar to the switch statement.

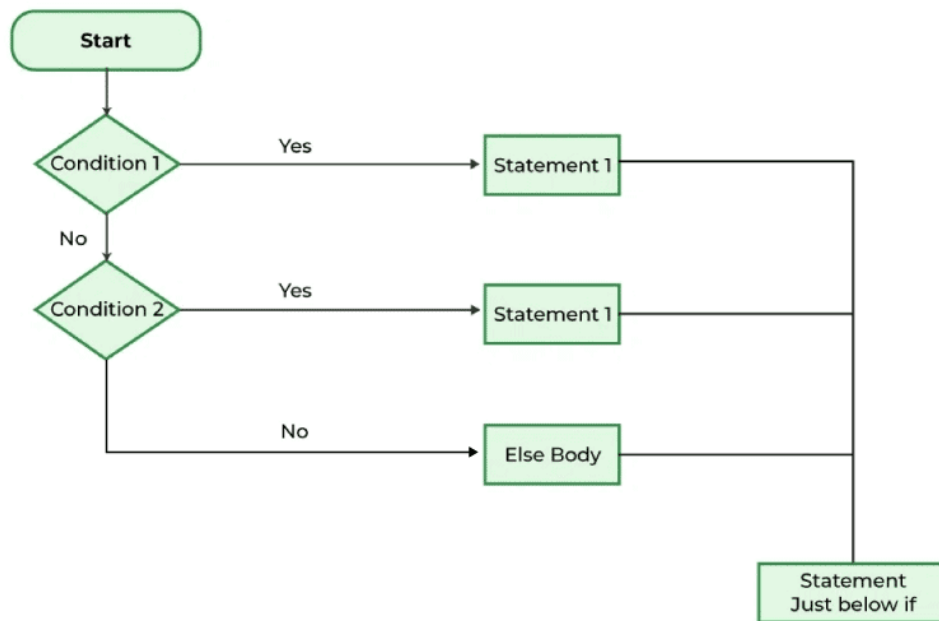
Syntax of if-else-if

```

if (condition)
    statement;
else if (condition)
    statement;
.
.
else
    statement;

```

Flowchart of if-else-if Ladder



Flow Diagram of if-else-

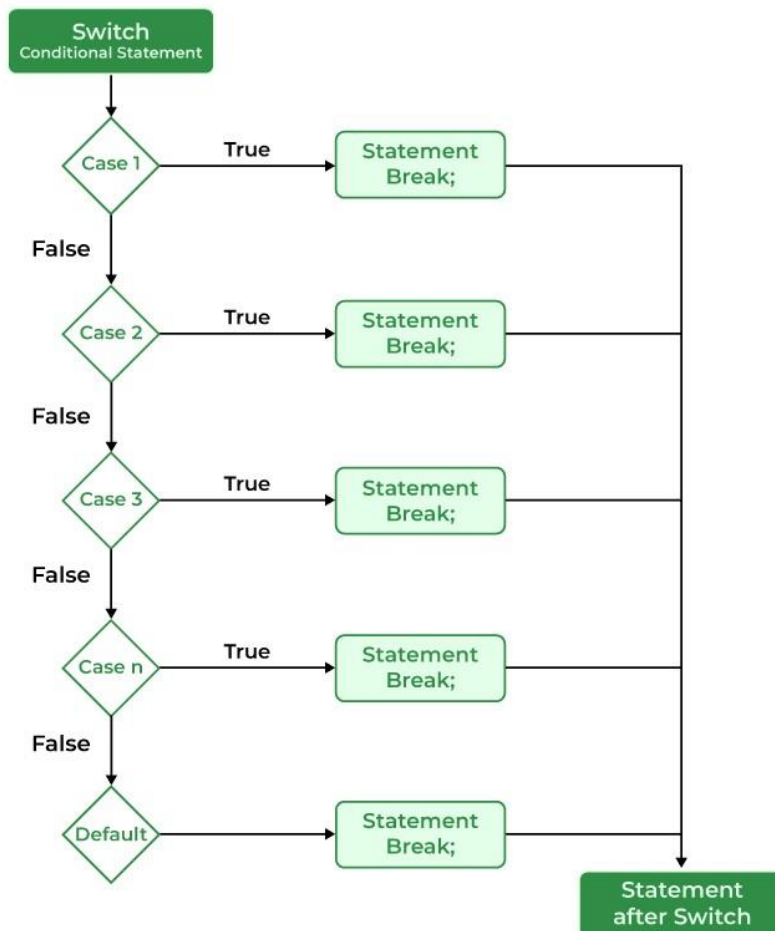
```
#include <stdio.h>
int main()
{
    int i = 20;
    if (i == 10)
    {
        printf("i is 10");
    }
    else if (i == 15)
    {
        printf("i is 15");
    }
    else if (i == 20)
    {
        printf("i is 20");
    }
    else
        printf("i is not present");
}
```

switch Statement in C

- The [switch case statement](#) is an alternative to the if else if ladder that can be used to execute the conditional code based on the value of the variable specified in the switch statement.
- The switch block consists of cases to be executed based on the value of the switch variable.

Syntax of switch

```
switch (expression) {  
    case value1:  
        statements;  
    case value2:  
        statements;  
    ....  
    ....  
    ....  
    default:  
        statements;  
}
```



```
#include <stdio.h>  
int main()  
{  
    // variable to be used in switch statement  
    int var = 2;  
  
    // declaring switch cases  
    switch (var) {  
        case 1:  
            printf("Case 1 is executed");  
            break;  
        case 2:
```

```

        printf("Case 2 is executed");
        break;
    default:
        printf("Default Case is executed");
        break;
    }

    return 0;
}

```

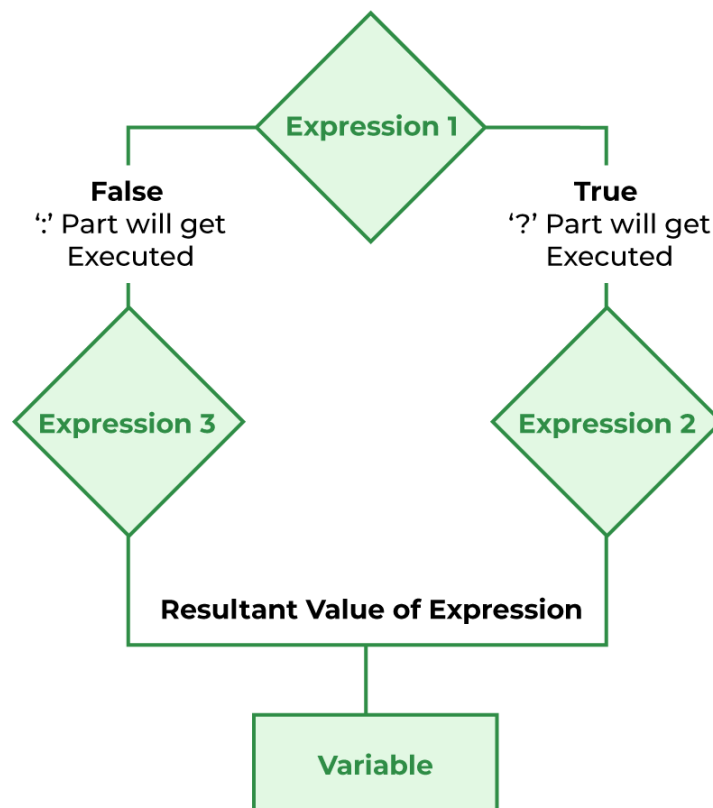
Conditional Operator in C

- The [conditional operator](#) is used to add conditional code in our program.
- It is similar to the if-else statement. It is also known as the ternary operator as it works on three operands.

Syntax of Conditional Operator

```
(condition) ? [true_statements] : [false_statements];
```

Flowchart of Conditional Operator



Example program

```
#include <stdio.h>

// driver code
int main()
{

    int var;
    int flag = 0;

    // using conditional operator to assign the value to var
    // according to the value of flag
    var = flag == 0 ? 25 : -25;
    printf("Value of var when flag is 0: %d\n", var);

    // changing the value of flag
    flag = 1;
    // again assigning the value to var using same statement
    var = flag == 0 ? 25 : -25;
    printf("Value of var when flag is NOT 0: %d", var);

    return 0;
}
```