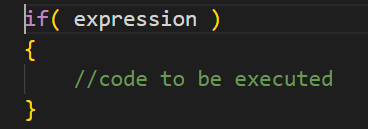
Decision making

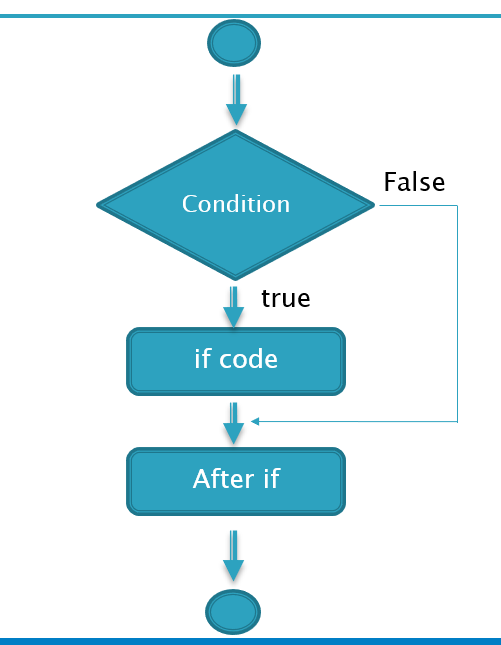
In C language, if-else and switch statements used to perform the operations based on some specific condition. The operations specific in if block are executed if and only if given conditions is true.

If statement:

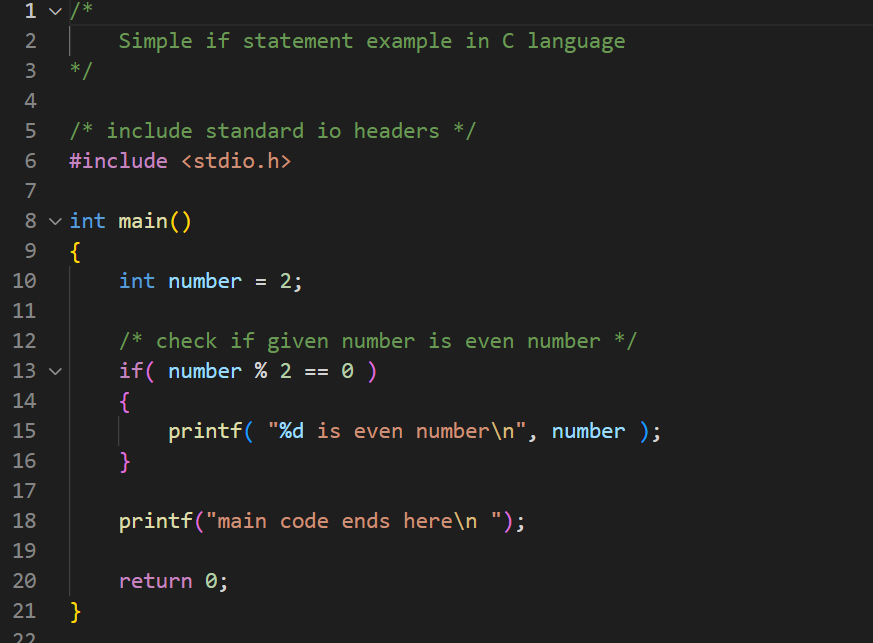
The if statement is used to check some given condition and perform some operations depending upon the correctness of that condition. It is mostly used in the scenario where we need to perform the different operations for the different conditions. The syntax of the if statement is given below.



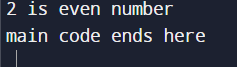
Flowchart of I statement



Lets see a simple example of if statement



Output



Program to find the largest number of three

/\* include standard header \*/

#include <stdio.h>

int main()

{

    int number\_1 = 6;

    int number\_2 = 45;

    int number\_3 = 25;

    if ( number\_1 > number\_2 && number\_1 > number\_3 )

    {

        printf( "%d is largest\n", number\_1 );

    }

    if (number\_2 > number\_1 && number\_2 > number\_3)

    {

        printf( "%d is largest\n", number\_2 );

    }

    if ( number\_3 > number\_1 && number\_3 > number\_2 )

    {

        printf( "%d is largest\n", number\_3 );

    }

    if ( number\_1 == number\_2 && number\_1 == number\_3 )

    {

        printf( "All are equal\n" );

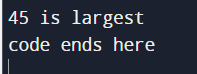
    }

    printf( "code ends here\n" );

    return 0;

}

Output



If-else statement

The if-else statement is used to perform two operations for a single condition. The if-else statement is an extension to the if statement using which, we can perform two different operations, i.e., one is for the correctness of that condition, and the other is for the incorrectness of the condition. Here, we must notice that if and else block cannot be executed simultaneously. Using if-else statement is always preferable since it always invokes an otherwise case with every if condition. The syntax of the if-else statement is given below.

#include <stdio.h>

if( expression )

{

    //code to be executed if condition is true

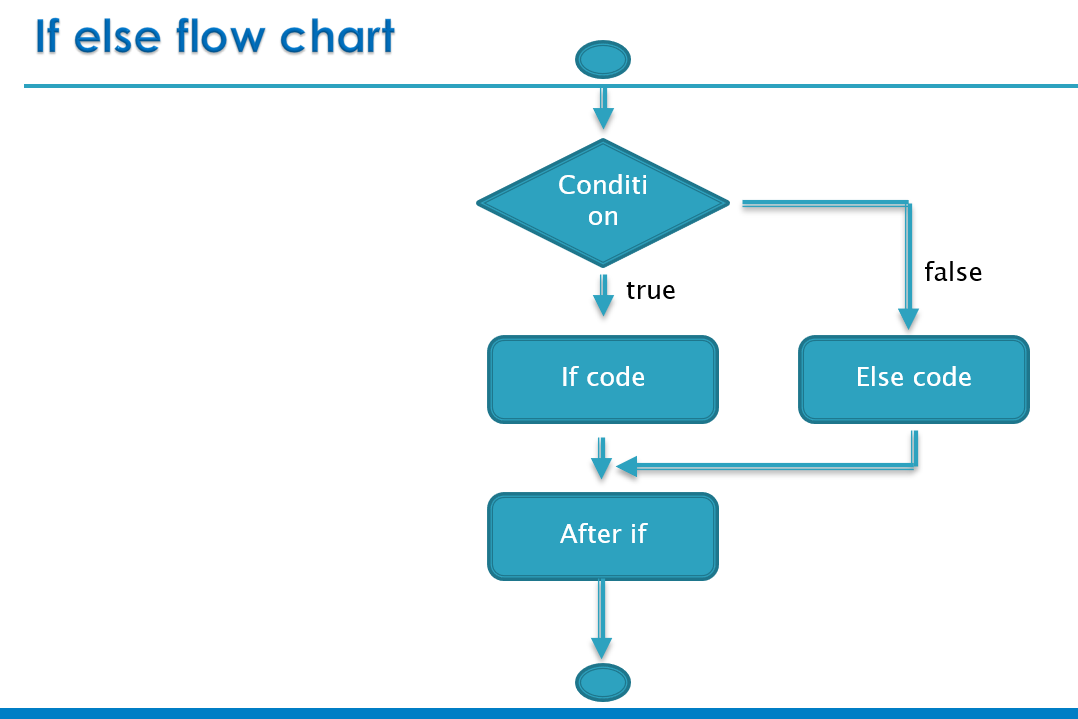
}

else

{

    //code to be executed if condition is false

}



Example code for if-else

/\*

    if else code example

\*/

/\* include standard headers \*/

#include <stdio.h>

int main()

{

    int number = 2;

    if ( number % 2 == 0 )

    {

        printf( "%d is even number\n", number );

    }

    else

    {

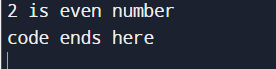
        printf( "%d is odd number\n", number );

    }

    printf( "code ends here\n" );

    return 0;

}



Program to check whether person is eligible to vote or not.

/\* include standard headers \*/

#include <stdio.h>

int main()

{

    int age = 15;

    if(age>=18)

    {

        printf("You are eligible to vote\n");

    }

    else

    {

        printf("Sorry ... you are not eligible to vote\n");

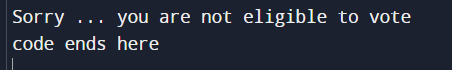
    }

    printf( "code ends here\n" );

    return 0;

}

Output



If else-if ladder statement

The if-else-if ladder statement is an extension to the if-else statement. It is used in the scenario where there are multiple cases to be performed for different conditions. In if-else-if ladder statement, if a condition is true then the statements defined in the if block will be executed, otherwise if some other condition is true then the statements defined in the else-if block will be executed, at the last if none of the condition is true then the statements defined in the else block will be executed. There are multiple else-if blocks possible. It is similar to the switch case statement where the default is executed instead of else block if none of the cases is matched.

#include <stdio.h>

if( condition\_1 )

{

    //code to be executed if condition\_1 is true

}

else if( condition\_2 )

{

    //code to be executed if condition\_2 is true

}

else if( condition\_3 )

{

    //code to be executed if condition\_3 is true

}

....

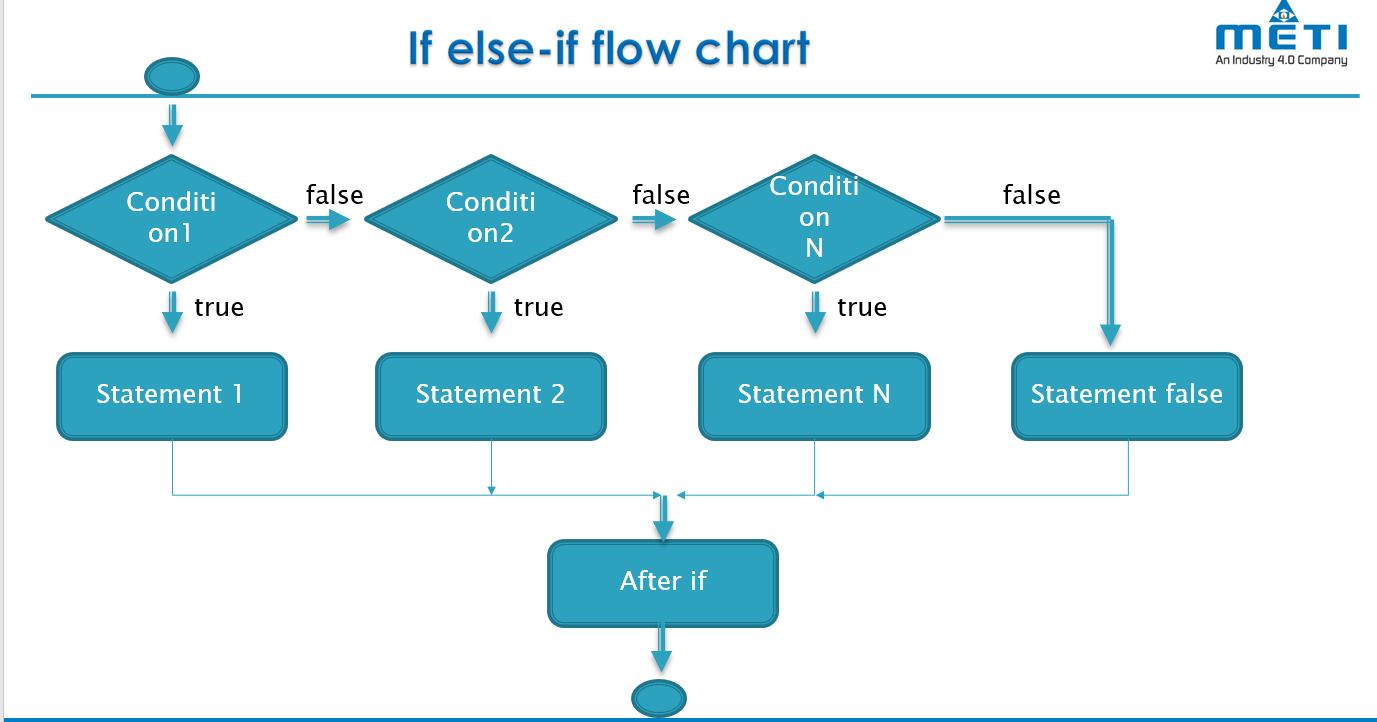
else

{

    //code to be executed if all conditions are false

}

Flow chart



If else-if example

/\*

    example code for if else-if statement

\*/

#include <stdio.h>

int main()

{

    int number = 66;

    if ( number == 10 )

    {

        printf( "number is equals to 10\n" );

    }

    else if ( number == 50 )

    {

        printf( "number is equal to 50\n" );

    }

    else if ( number == 100 )

    {

        printf( "number is equal to 100\n" );

    }

    else

    {

        printf( "number is not equal to 10, 50 or 100\n" );

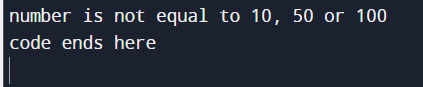
    }

    printf( "code ends here\n" );

    return 0;

}

Output:



Program to calculate grades of students

/\*

    example code to print grades of students

\*/

#include <stdio.h>

int main()

{

    int marks = 64;

    if (marks > 85 && marks <= 100)

    {

        printf("Congrats ! you scored grade A ...\n");

    }

    else if (marks > 60 && marks <= 85)

    {

        printf("You scored grade B + ...\n");

    }

    else if (marks > 40 && marks <= 60)

    {

        printf("You scored grade B ...\n");

    }

    else if (marks > 30 && marks <= 40)

    {

        printf("You scored grade C ...\n");

    }

    else

    {

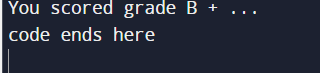
        printf("Sorry you are fail ...\n");

    }

    printf( "code ends here\n");

    return 0;

}



Switch statement

The switch statement in C is an alternate to if-else-if ladder statement which allows us to execute multiple operations for the different possible values of a single variable called switch variable. Here, We can define various statements in the multiple cases for the different values of a single variable.

#include <stdio.h>

switch( expression )

{

    case value1:

    //code to be executed;

    break;  //optional

    case value2:

    //code to be executed;

    break;  //optional

    ......

    case valuen:

    //code to be executed;

    break;  //optional

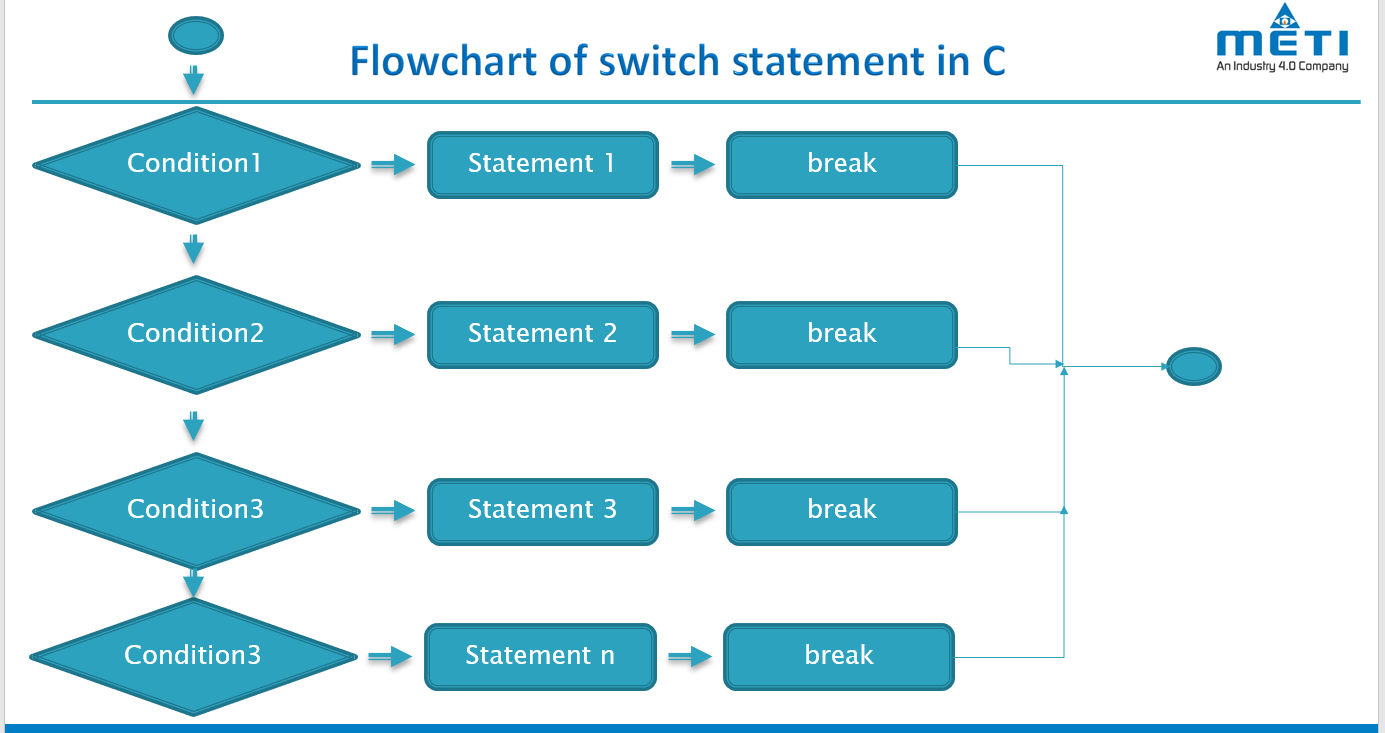
    default:

    //code to be executed if all cases are not matched;

    break;

}

Flowchart for switch statement



Simple example

/\* switch statement example \*/

#include <stdio.h>

int main()

{

    int number = 65;

    switch (number)

    {

        case 10:

            printf("number is equals to 10\n");

            break;

        case 50:

            printf("number is equal to 50\n");

            break;

        case 100:

            printf("number is equal to 100\n");

            break;

        default:

            printf("number is not equal to 10, 50 or 100\n");

            break;

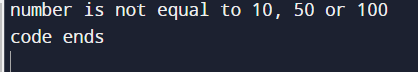
    }

    printf("code ends\n");

    return 0;

}

Output:



Nested switch statement example

/\* nested switch example \*/

#include <stdio.h>

int main()

{

    int i = 10;

    int j = 20;

    switch (i)

    {

        case 10:

            printf("the value of i evaluated in outer switch: %d\n", i);

        case 20:

            switch (j)

            {

                case 20:

                printf("The value of j evaluated in nested switch: %d\n", j);

            }

    }

    printf("Exact value of i is : %d\n", i);

    printf("Exact value of j is : %d\n", j);

    printf("code ends here \n");

    return 0;

}

output

