

Programming and Problem Solving through C Language O Level / A Level

Chapter -4 : Conditional Statements and Loops

Decision Making within a Program

- Decision making is the selection of a course of action from among available alternatives in order to produce a desired result.
- The conditional test either evaluates to a true or a false.
- The concept of evaluating and obtaining a result is referred to as decision making in a programming language.
- "True" is considered the same as "yes," which is also considered the same as 1.
- "False" is considered the same as "no," which is considered the same as 0.
- C programming language assumes any non-zero and non-null values as true, and if it is either zero or null, then it is assumed as false value.

Control Statement

- A control statement modifies the order of statement execution.
- A control statement can cause other program statements to execute multiple times or not to execute at all, depending on the circumstances.

Types of Control Statement

1) **Branching Statement** : used to select one of the alternative statement

a. Unconditional Branching

- i. Goto Statement

b. Conditional Branching

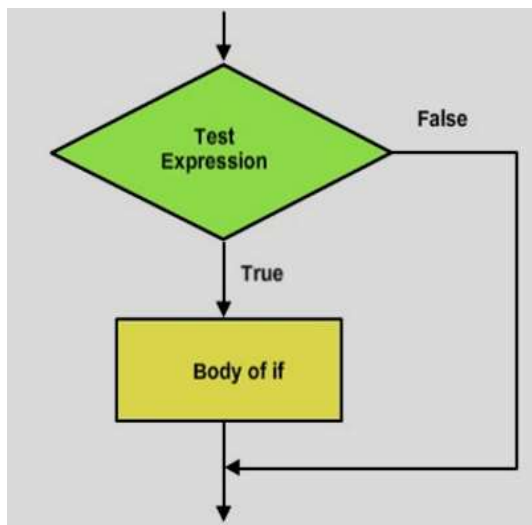
- i. if State
- ii. if-else Statement.
- iii. switch Statement.

2) **Looping or Iterative Statement** : used to repeat the statement till the condition is true.

- i. for loop
- ii. while loop
- iii. do while loop

If statement

- The 'if' statement is one of the C's program control statements.
- The 'if' statement evaluates an expression and directs program execution depending on the result of the evaluation.
- If expression evaluates to true, statement is executed.
- If statement evaluates to false, statement is not executed.
- An 'if' statement can control the execution of multiple statements through the use of a compound statement, or block.
- A block is a group of two or more statements enclosed in braces.



If statement Syntax

```
..  
if (condition)  
{  
    //Block of C statements here  
    //The above statements will only execute if the condition is true  
}  
..
```

If statement: Example

```
#include <stdio.h>  
int main()  
{  
    int x = 20;  
    int y = 22;  
    if (x<y)  
    {  
        printf("Variable x is less than y");  
    }  
    return 0;  
}
```

Output:

Variable x is less than y

Else Statement

- An 'if' statement can optionally include an else clause.
- The else clause is included as shown below:
- If expression evaluates to true, statement1 is executed.
- If expression evaluates to false, statement2 is executed.
- Both statement1 and statement2 can be compound statements or block

If-else Statement

- The combination of the 'if' and 'else' clause is called the 'if-else' statement.
- If expression is true, statement1 is executed; otherwise, statement2 is executed.
- If the first expression, expression1, is true, statement1 is executed before the program continues with the next statement.

If-else statement Syntax

```
if (condition)
{
    /*Control will come inside only when the above condition is true*/
    //C statement(s)
}
else
{
    /*Control will come inside only when condition is false */
    //C statement(s)
}
```

```
#include <stdio.h>
int main()
{
    int m=40, n=20;
    if(m == n)
    {
        printf("m and n are equal");
    }
    else
    {
        printf("m and n are not equal");
    }
}
```

Output:

m and n are not equal

```

#include <stdio.h>
int main()
{
    int m=40, n=20;
    int o=20, p=30;
    if(m>n && m!=0)
    {
        printf("&& Operator: Both conditions are true\n");
    }
    if(o>p || p!=20)
    {
        printf("|| Operator: Only one condition is true\n");
    }
    if(!(m>n && m!=0))
    {
        printf("! Operator: Both conditions are true\n");
    }
    else
    {
        printf("! Operator: Both conditions are true. " \
            "But, status is inverted as false\n");
    }
}

```

Output

```

&& Operator: Both conditions are true
|| Operator: Only one condition is true
! Operator: Both conditions are true. But, status is inverted as false

```

Assignment

- 1) Write a program to display the square and cube of a positive number.
- 2) Write a program to display the greater of 2 numbers.
- 3) Write a program to check an entered number is Odd or Even. [hint – use % modulus operator to determine the remainder]
- 4) Write a program to check an entered number is divisible by 7 or not.
- 5) In an examination, the grades are awarded to the students in 'SCIENCE' according to the average marks obtained in the examination.

Marks	Grades
80% and above	Distinction
60% or more but less than 80%	First Division
45% or more but less than 60%	Second Division
40% or more but less than 45%	Pass
Less than 40%	Promotion not granted

Write a program to input marks in Physics , Chemistry and Biology. Calculate the average marks. Display the average marks and grade obtained.