Control Structure

It is a part of structure program which includes:

a. Condition
b. Code Block

If the condition is – (a) TRUE: Runs the code block

(b) FALSE: Exit

Two types of Control structure:

1. Non-iterative:
   Runs one time if the condition is true.

   a. Two-way: if-else
   b. Multi-way: switch

2. n-iterative:
   Repeats the same code block accordingly till the condition does not get to be FALSE.

   a. Entry controlled: for loop, while loop
   b. Exit controlled: do while loop

If-else

- It is a two way non-iterative control structure.
- Syntax:

```
if(condition){
    //code block;
}
```

  If the condition is true, the code block gets executed, otherwise not.
- If the code block contains only one statement, then { } is not necessary.
- If no condition is given, default condition is (!=0).
  Example: if(5) means if(5!=0).

Example for if-else:

```
#include<stdio.h>

int main(){
    if(4>2)
        printf("True");
    else
        printf("False");
    return 0;
}
```

```
True
```

If-else ladder:

It is used when multiple conditions controlled different output.

Syntax:

```
if(condition 1){
    //codeblock 1
}
else if(condition 2){
    //codeblock 2
}
else if(condition 3){
    //codeblock 3
}
.
.
else{
    //codeblock n
}
```

Here, if the condition 1 is true, then code block 1 gets executed and the program exits itself. On the other hand, if it is false, the program checks if condition 2 is true, and so on.

If any of the conditions are not true, then the last else code block gets executed.

Note: One can use ternary operator (Conditional operator) instead of if-else.

For example:

```
if(4>2)
        printf("True");
else
        printf("False");
```

Above code can also be written as –

```
(4>2)?printf("True"):printf("False");
```

Example for if-else ladder:

```
//check which is greater between a,b,c
int main(){
    int a,b,c;
    printf("Enter 3 numbers: ");
    scanf("%d%d%d",&a,&b,&c);
    if(a>b&&a>c)
        printf("a is greatest");
    else if(b>c)
        printf("b is greatest");
    else
        printf("c is greatest");
}
```

```
Enter 3 numbers: 12  13  14
c is greatest.

Enter 3 numbers: 43  112  54
b is greatest.

Enter 3 numbers: 12 5 8
a is greatest.
```

Nested if-else:

It is used when one condition is depends upon other condition.

Syntax:

```
if(condition 1){
    if(condition 2){
        if(condition 3){
            //codeblock
        }
    }
}
```

Here, if all the three conditions i.e. condition 1, condition 2 and condition 3 are true, then only the code block gets executed otherwise not.

Example for nested if else:

```
// check if a character is alphabet or a digit
#include<stdio.h>
#include<conio.h>
#include<ctype.h>
int main(){
    char input;
    printf("Enter a character: ");
    input = getche();
    if(isalnum(input)){
        if(isalpha(input)){
            printf("%c is an alphabet",input);
        }
        else{
            printf("%c is a digit",input);
        }
    }
    return 0;
}
```

```
Enter a character: e
e is an alphabet

Enter a character: 3
3 is an digit
```

Here, if a character is alphabet or a digit then only the program starts to execute the next if condition. Functions isalnum(), isalpha() are included inside the header file ctype.h.

Switch-case

- It is a multi-way non-iterative control structure.
- The control variable can be of any data type and the switch works on same variable.
- Unlike if-else ladder, switch jumps the program control directly to the case with matching value.
- If no matching case is defined, the default statement works.
- The cases do not need to be in any order, they can be in any order.

- Syntax:

```
switch(ch){
        case 1: //statement 1
                break;
        case 2: //statement 2
                break;
        .
        .
        case n: //statement n
                break;
        default: //default statement
}
```

Here, **ch** is the control variable onto whose value, the control structure works. If the value of **ch** is 1, then statement 1 works; if the value of **ch** is 2, then statement 2 works and so on. If the value of **ch** does not matches any defined case value, then the default statement executes.

Note: The *break* keyword is used to exit from the switch control structure when any of the statement gets executed. If we do not write *break* keyword, every subsequent case will get executed i.e. , if the value of **ch** is 2 and no *break* is written then statement 2, statement 3 and every other statements written after case 2 will get executed.

Example of switch-case:

```
#include<stdio.h>
int main(){
    int ch;
    printf("Enter a number between 1 and 5: ");
    scanf("%d",&ch);
    switch(ch){
        case 1: printf("You have chosen 1");
                break;
        case 2: printf("You have chosen 2");
                break;
        case 3: printf("You have chosen 3");
                break;
        case 4: printf("You have chosen 4");
                break;
        case 5: printf("You have chosen 5");
                break;
        default:printf("Wrong value");
    }
    return 0;
}
```

```
Enter a number between 1 and 5: 2
You have chosen 2

Enter a number between 1 and 5: 4
You have chosen 4

Enter a number between 1 and 5: 8
Wrong value
```