# Programming Fundamentals Lab

Lab # 08

Selection Control Structure (if, if else, multiple if, multiple if else if), scanf() function

Instructor: Fariba Laiq

Email: fariba.laiq@nu.edu.pk

Course Code: CL1002

Semester Fall 2022

Department of Computer Science,

National University of Computer and Emerging Sciences FAST
Peshawar Campus

# Contents

## The scanf() function

In C, scanf() is the most commonly used function to take input from user from the standard input device that is keyboard.

**Syntax:**

scanf(format specifier, &variable_name);

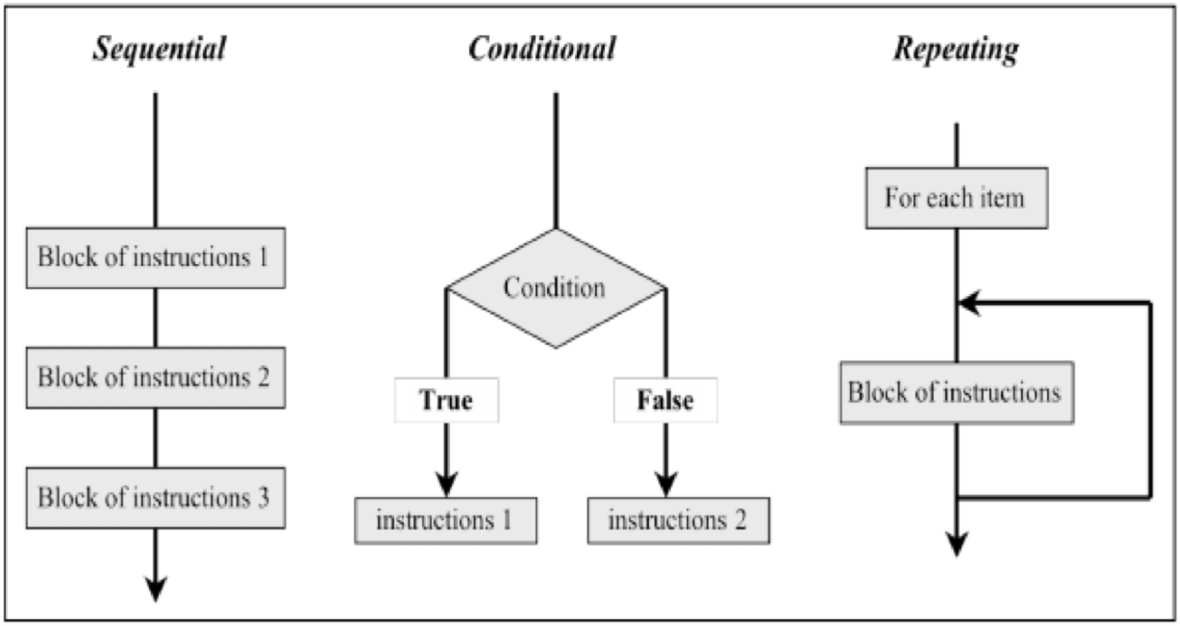
**Eg:**

scanf(“%d”, &age);

Here %d is the format specifier for int data type which means that we are taking an int type variable from user. The & is the reference operator which gives the address of a variable. Here &age means to store the input data on the address of age variable.

## Control Structures in C

Algorithms require two important control structures: iteration(repeating) and selection(conditional). Both are supported by C in various forms. The programmer can choose the statement that is most useful for the given circumstance.

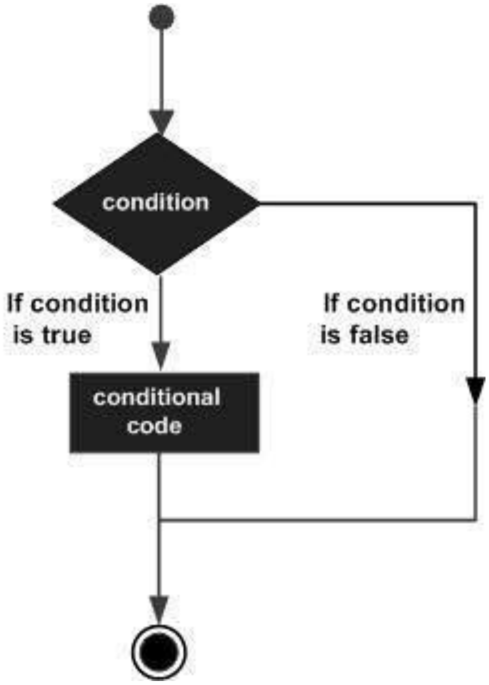


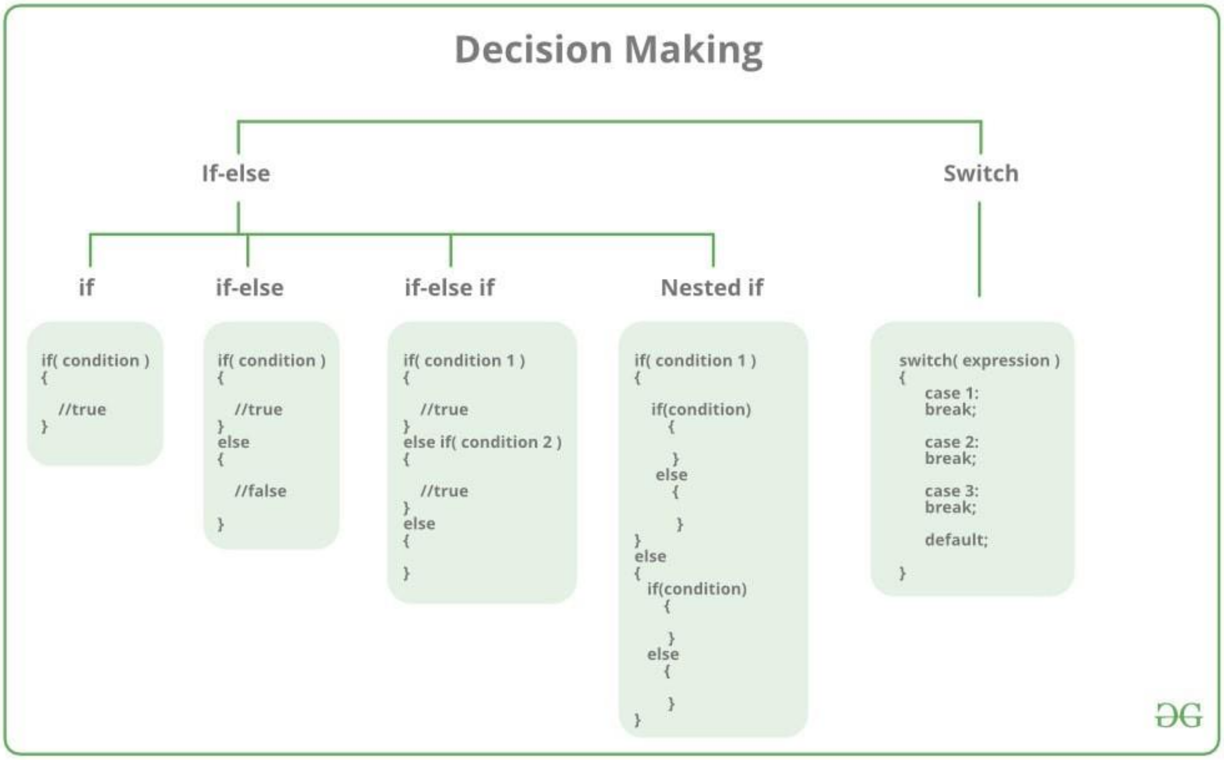
## Selection Statements / Decision Making

- There come situations in real life when we need to make some decisions and based on these decisions, we decide what should we do next. Similar situations arise in programming also where we need to make some decisions and based on these decisions, we will execute the next block of code.
- For example, in C if x occurs then execute y else execute z. There can also be multiple conditions like in C if x occurs then execute p, else if condition y occurs execute q, else execute r. This condition of C else-if is one of the many ways of importing multiple conditions.
- Decision making is the most important aspect of almost all the programming languages. As the name implies, decision making allows us to run a particular block of code for a

particular decision. Here, the decisions are made on the validity of the particular conditions. Condition checking is the backbone of decision making.

- Following is the general form of a typical decision-making structure found in most of the programming languages.





```
if( condition )
{
   //true
}
```

```
if( condition )
{
   //true
}
else
{
   //false
}
```

```
if( condition 1 )
{
   //true
}
else if( condition 2 )
{
   //true
}
else
{
}
```

```
if( condition 1 )
{
   if(condition)
      {
      }
    else
      {
      }
}
else
{
  if(condition)
     {
     }
  else
     {
     }
}
```

```
switch( expression )
{
    case 1:
    break;

    case 2:
    break;

    case 3:
    break;

    default;
}
```

## C if else Statement

The if-else statement in C is used to perform the operations based on some specific condition. The operations specified in if block are executed if and only if the given condition is true.

There are the following variants of if statement in C language.

- If statement
- If-else statement
- If else-if ladder
- Nested if/ Nested if-else

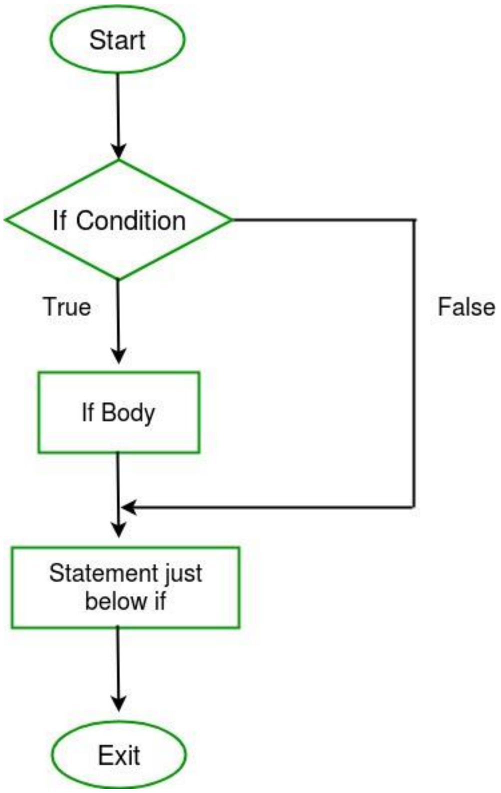
## 1. If Statement in C (One-Way Decision)

if statement is the simplest decision-making statement. It is used to decide whether a certain statement or block of statements will be executed or not i.e if a certain condition is true then a block of statement is executed otherwise not.

**Syntax:**

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

## Flowchart of if statement



**Example 1**

```
// C program to illustrate If statement

#include <stdio.h>
  int main()
{
      int i;
       printf("Enter an integer: ");
       scanf("%d", &i);
      if (i > 0) {
        printf("\ni is positive ");
      }
}
```

**Output**

```
Enter an integer: 4

i is positive
```

If the condition present in the if statement is false. So, the block below the if statement is not executed.

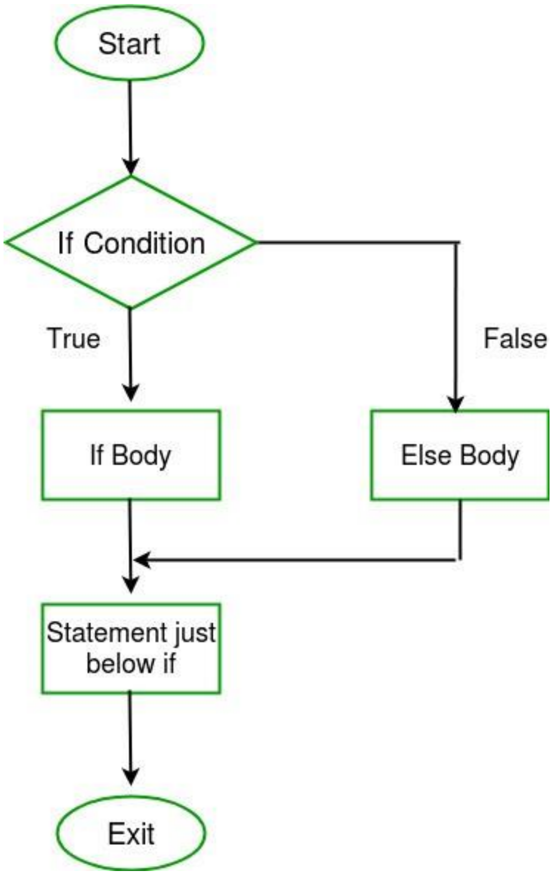
## 2. if-else in C (Two-Way Decision)

The if statement alone tells us that if a condition is true it will execute a block of statements and if the condition is false it won’t. But what if we want to do something else if the condition is false. Here comes the C else statement. We can use the else statement with the if statement to execute a block of code when the condition is false.

**Syntax**:

```
if (condition)

{

   // Executes this block if

   // condition is true

}

else

{

   // Executes this block if

   // condition is false

}
```

Flowchart of if-else



**Example 2**

```
#include <stdio.h>

int main()
{
        int i = 20;

        if (i >= 0) {

                printf("i is positive or zero");
        }
        else {

                printf("i is negative");
        }
return 0;
}
```

**Output**

```
i is positive or zero
```

The block of code following the else statement is executed as the condition present in the if statement is false.

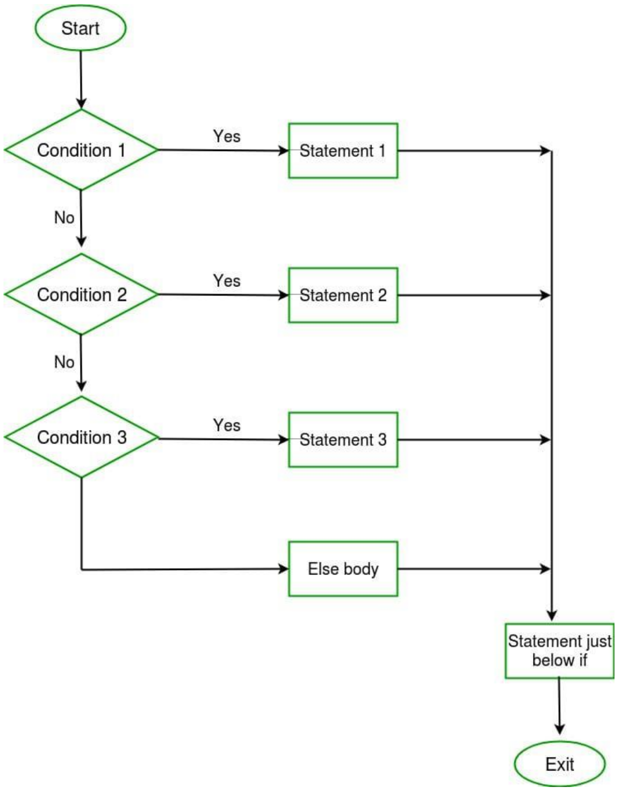
## 3. if-else-if ladder in C (Multi Way Decisions)

Here, a user can 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.

**Syntax:**

```
if (condition)
{
     Statements;
}
else if (condition)
{
     Statements;
}
else
{
      Statements to execute if   none of the above
      conditions are true;
}
```

## Flowchart of if-else-if



**Example 3:**

```
#include <stdio.h>
int main()
{
        int i = -10;
        if (i > 10){
                printf("i is Positive");
        }
        else if (i == 0){
                printf("i is zero");
        }
        else{
                printf("i is negative");
        }
}
```

Output:

```
i is negative
```

## C Logical Operators

An expression containing logical operator returns either 0 or 1 depending upon whether expression results true or false. Logical operators are commonly used in decision making in C programming.

Example 4:

```
#include <stdio.h>
int main()
{
int age = 11;
if(age>=13 && age<=19)
{
   printf("You are teenager!\n");
}
else
{
   printf("you are not a teenager\n");
}
return 0;
}
```

Output

```
You are teenager!
```

**Example 5**

```
#include <stdio.h>
int main()
{
        char c;
        printf("Enter a char:");
        scanf("%c", &c);
        if(c=='a')
        {
                printf("\nYou entered the first letter of abc....");
        }
        else if(c=='z')
        {
                printf("\nYou entered the last letter of abc....");
        }
        else
        {
                printf("\nYou entered any number other than a or z");
        }
return 0;
}
```

Output

```
Enter a char: a

You entered the first letter of abc....
```

## References:

*https://www.geeksforgeeks.org/decision-making-c-cpp/*
*https://www.javatpoint.com/c-if-else*