Parul University

Fundamentals of Programming Using C - 15101104

Unit 2 : Pre-Processor, Storage Classes, Conditional Statement and Loop in C

Bhavika Vaghela

Assistant Professor

Parul Institute of Computer Application

Faculty of IT & Computer Science

Parul University

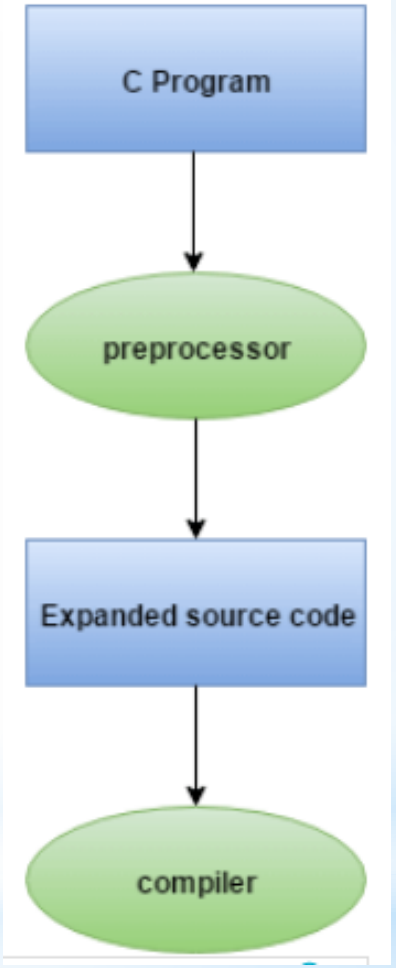
Pre-Processor

- Pre-Processor is a part of compiler it is use to compile your C program. Which allow you to use macro.
- A Preprocessor is a system software (a computer program that is designed to run on computer's hardware and application programs).
- In simple terms, a C Preprocessor is just a text substitution tool and it instructs the compiler to do required pre-processing before the actual compilation
- We'll refer to the C Preprocessor as CPP.
- All preprocessor commands begin with a hash symbol (#).
- Preprocessor directives are executed before compilation

Cont…

#include
#define
#undef
#ifdef
#ifndef
#if
#else
#elif
#endif
#error
#pragma

Example of Preprocessor



Macro

What is Macro?

- A macro is segment of code which is declare using #define with some value like, #define PI 3.14.

We have two types of macro

- Object like macro
- Function like macro

The **object-like macro** is an identifier that is replaced by value. It is widely used to represent numeric constants.

- Function-like Macros: The function-like macro looks like function call. For example:

```
#define MIN(a,b) ((a)<(b)?(a):(b))
```

Here, MIN is the macro name.

Storage Classes

- A storage class is used to represent additional information about a variable.
- Storage class represents the scope and lifespan of a variable.
- It also tells who can access a variable and from where?
- **Auto, extern, register, static** are the four storage classes in 'C'.
- **auto** is used for a local variable defined within a block or function
- **register** is used to store the variable in CPU registers rather memory location for quick access.
- **Static** is used for both global and local variables. Each one has its use case within a C program.
- **Extern** is used for data sharing between C project files.

Cont…

- Please refer unit 1 for static, global and auto storage class in topic types of variable.

Let's discuss about register storage class.

- Variable using register storage declare same as other variable type.
- For that **register** keyword is use like,

register int num=10; or register int num;

- It will store variable in register and not into memory because accessing value from register become fast rather than memory.
- If register is free than only it occupy space over there other wise not.
- It can't have the unary '&' operator applied to it (as it does not have a memory location).

Decision Making and Control Statement

- C provides two styles of flow control:
  - Branching
  - Looping
- Branching is deciding what actions to take and looping is deciding how many times to take a certain action.
- In Branching : if, if… else, if… else i…. else and nested if is available. (it will execute based on the given condition)
- In Looping : for loop, while loop and do while loop is available, it is use to execute same task n numbers of time.
- One more Decision making statement provide as switch…case.
- Lets discuss it in detials.

Decision Making (statement)

If statement : if given condition become true than and than the inner block is going to execute. Other wise not.

Syntax :

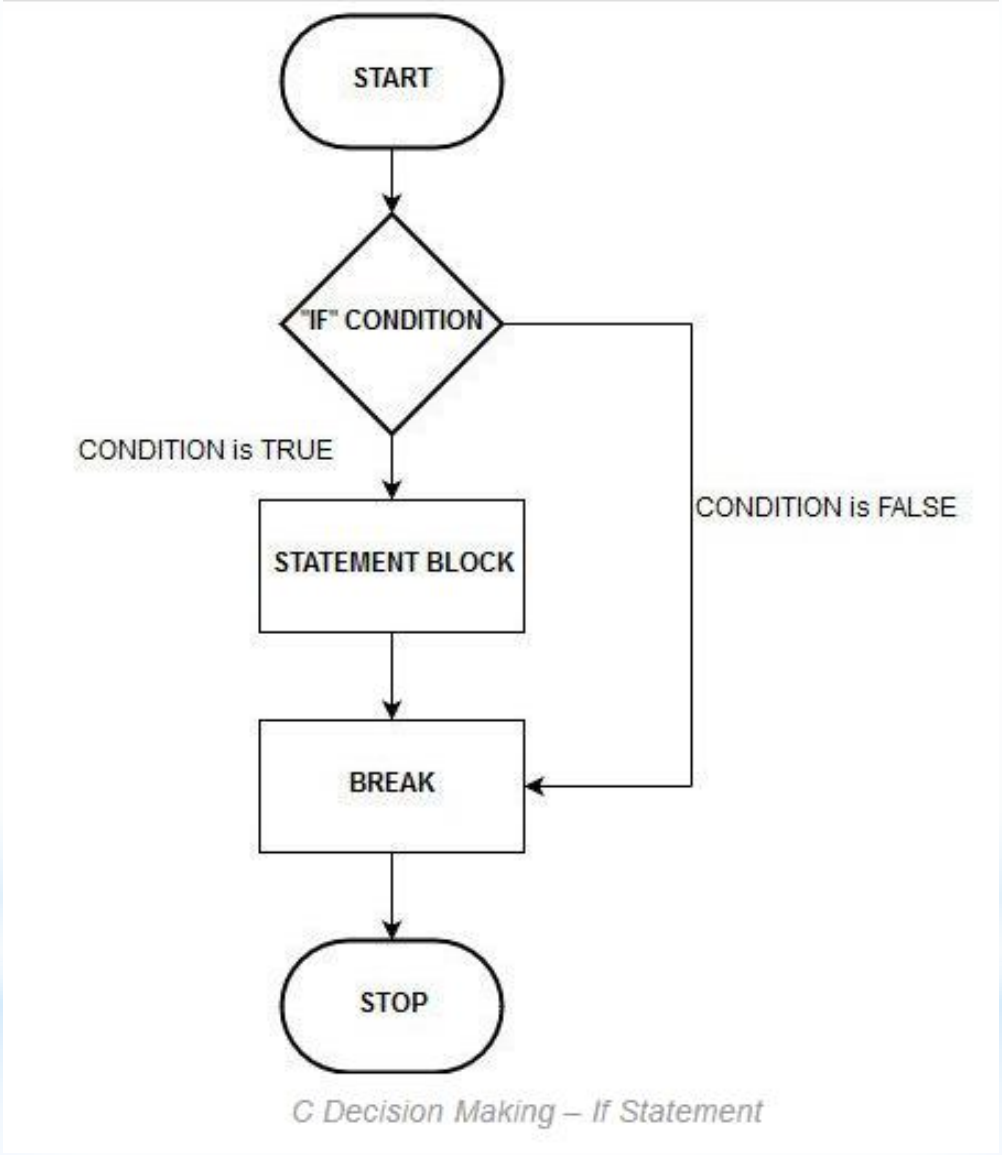
```
If (condition)
{
    //executable statments
}
```

We can declare more than one if statement within a single program as per requirement.

Program

```
#include<conio.h>
#include<stdio.h>
void main()
{
    int num=10;
    clrscr();
    if(num>0)
    {
       printf("positive number");
    }
    getch();
}
```

Cont…


START
"IF" CONDITION
CONDITION is TRUE
CONDITION is FALSE
STATEMENT BLOCK
BREAK
STOP
C Decision Making – If Statement

Program using if

```
//Program using more than one if statement
#include<stdio.h>
#include<conio.h>
void main()
{
        int num=10;
        clrscr();
        if(num>0)
        {
                printf("\n Number is postive");
        }
        if(num<0)
        {
                printf("\n Number is negative");
        }
        if(num%2==0)
        {
                printf("\n number is even and divisible by 2");
        }
        if(num%3==0)
        {
                printf("\n number is divisible by 3");
        }
        getch();

}
```

Cont…

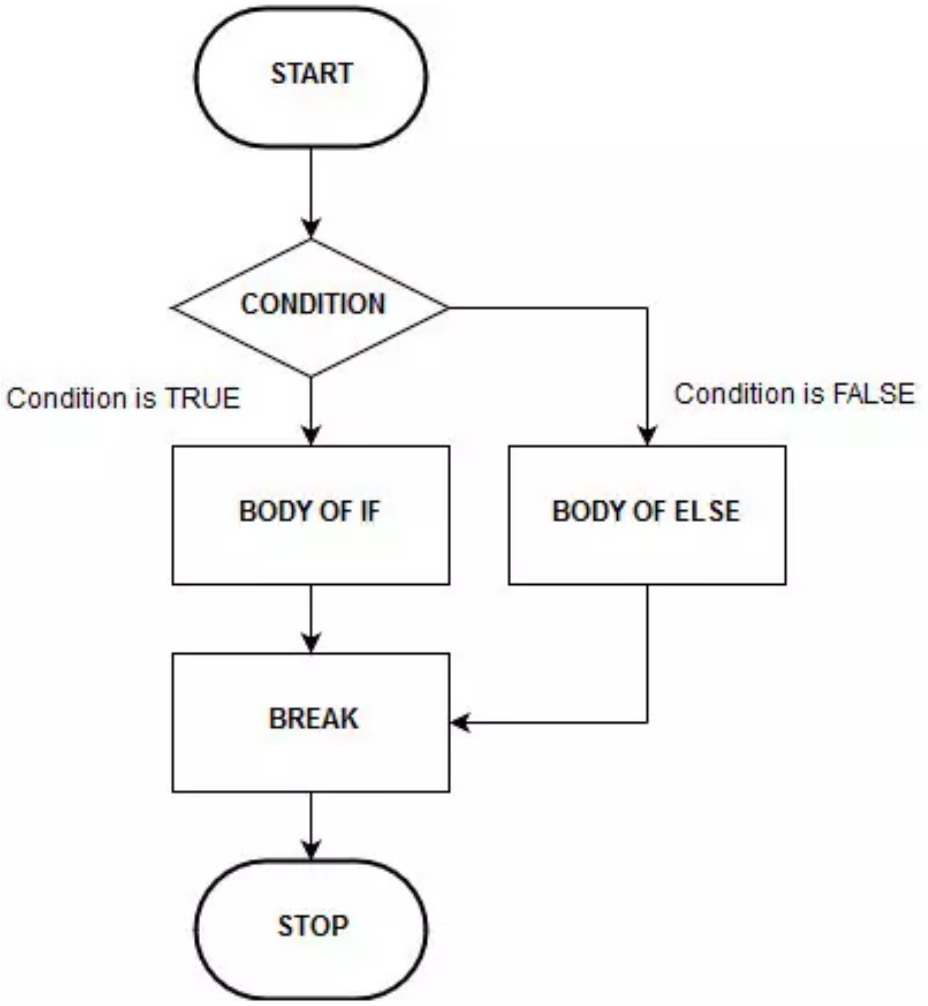
if….else statement

- if else is use to perform two operation using single statement.
- In if else statement else part is know as branch of it.
- If the condition become true than inner block of if statement is executed other wise branch (else block) is going to executed.

```
Syntax :
If(condition)
{
        executable statements;
}
else
{
        executable statements;
}
```

Cont…

Flow chart of if … else statement



C Decision Making – If…Else Statement

Note : you can also declare more than one if else statements in single program.

Program using if…else

```
#include<conio.h>
#include<stdio.h>
void main()
{
        int num=30;
        clrscr();
        if(num%3==0 && num%2==0)
        {
                printf("\n Number is divisible by 3 and 2 and its even number");
        }
        else
        {
                printf("\n number is note divisible by 3 and 2");
        }
        getch();
}
```

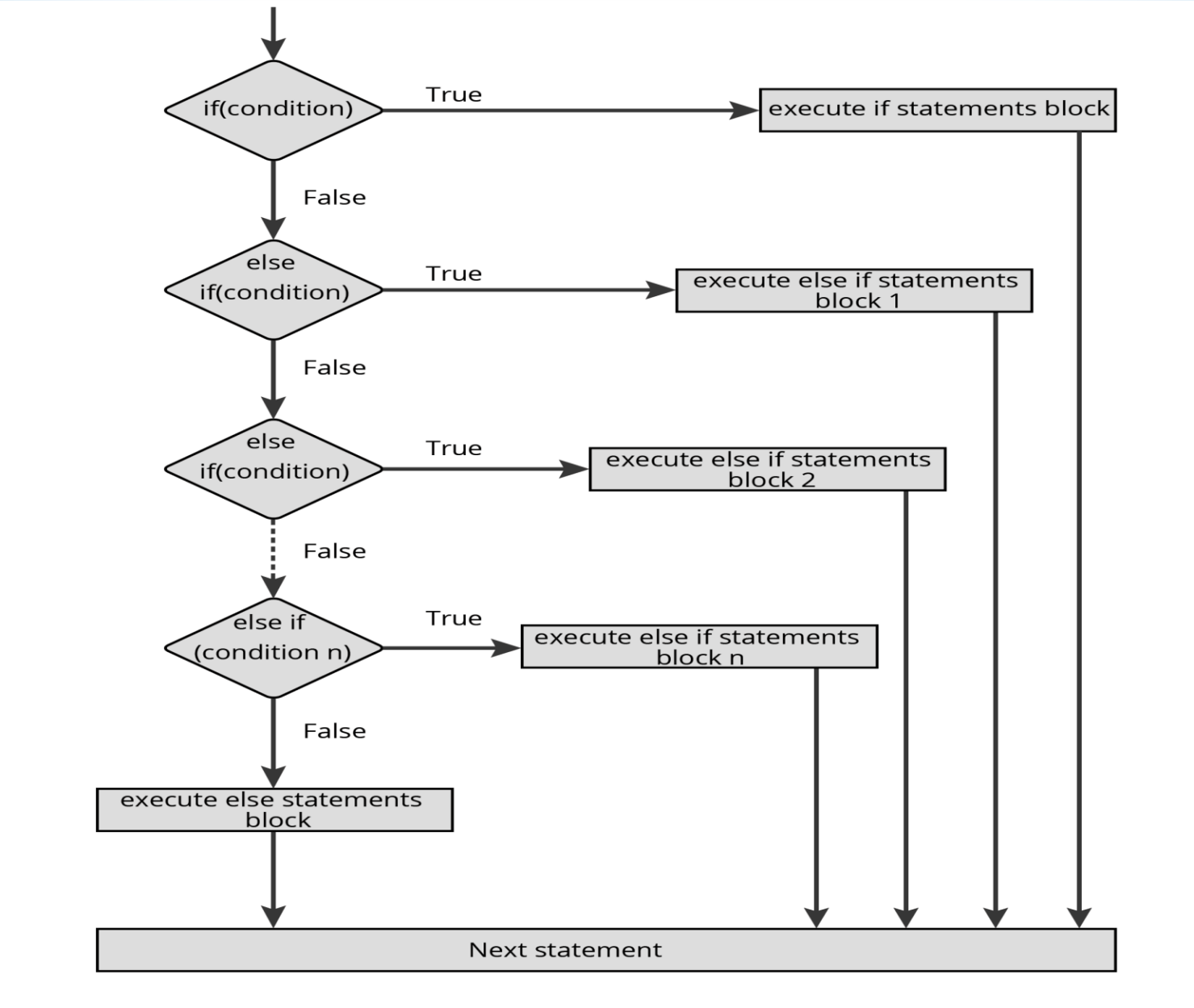
if else if else (if else ladder statement)

It execute two different code based upon the condition it going to execute the related branch.

It must have else statement at the last. Please check syntax below.

```
If(condition 1)
{   //statement 1 }
else if (condition 2)
{   //statement 2 }
else if (condition 3)
{   //statement 3 }
else if (condition 4)
{   //statement 4 }
else if (condition 5)
{   //statement 5 }
…….
…….
…….
else
{    //statement n}
```

Flow chart of if else if else



Program using if else if else

```
//Program to show relation between two numbers using if else if else
#include<conio.h>
#include<stdio.h>
int main()
{
        int num1,num2;
        clrscr();
        printf("\n enter value of num1 : ");
        scanf("%d",&num1);
        printf("\n enter value of num2 : ");
        scanf("%d",&num2);
        if(num1==num2)
        {
                printf("\n num1 and num2 both are same");
        }
        else if(num1>num2)
        {
                printf("\n num1 is bigger than num2 ");
        }
        else if(num1<num2)
        {
                printf("\n num1 is smaller than num2");
        }
        else if(num1==0 && num2==0)
        {
                printf("\n both number are ZERO");
        }
        else
        {
                printf("\n value of number is %d and number2 is %d",num1,num2);
        }
        getch();
        return 0;
}
```

Nested if statement

- One can also declare if within if to check more than one condition.
- It will execute like if condition1 become true than only it going to check inner condition else not.
- It must have one else statement. If not than it not give any syntax error but yes logical error will be there.

Syntax :

```
if(condition)
{   if(condition)
     {  if (condition)
        { if(condition)   …….. More if statements if required}
     }
}
else
{
     //statements
}
```

Program using nested if

```
#include <stdio.h>
int main()
{
   int number1, number2;
   printf("Enter two integers: ");
   scanf("%d %d", &number1, &number2);
   if (number1 >= number2)
   {  if (number1 == number2)
      {
           printf("Result:  %d = %d",number1,number2);
      }
      else
      {
            printf("Result:  %d > %d", number1, number2);
       }
    }
    else
    {
             printf("Result:  %d < %d",number1, number2);
    }
    return 0;
}
```

Switch Case statement

- Using switch case we can check more than one alternatives.
- You can also do the same using if else if else (ladder) statement.
- A switch statement is a conditional statement used in C programming to check the value of a variable and compare it with all the cases.
- It matches with all the alternatives and if it match than that particular block is going to executed. And other are not.
- If value enter by user not match with any case than it will executed default block. So default block is mandatory.
- And each block must have break statement at the end of block.
- We can also declare nested switch case.

Cont...

Syntax :

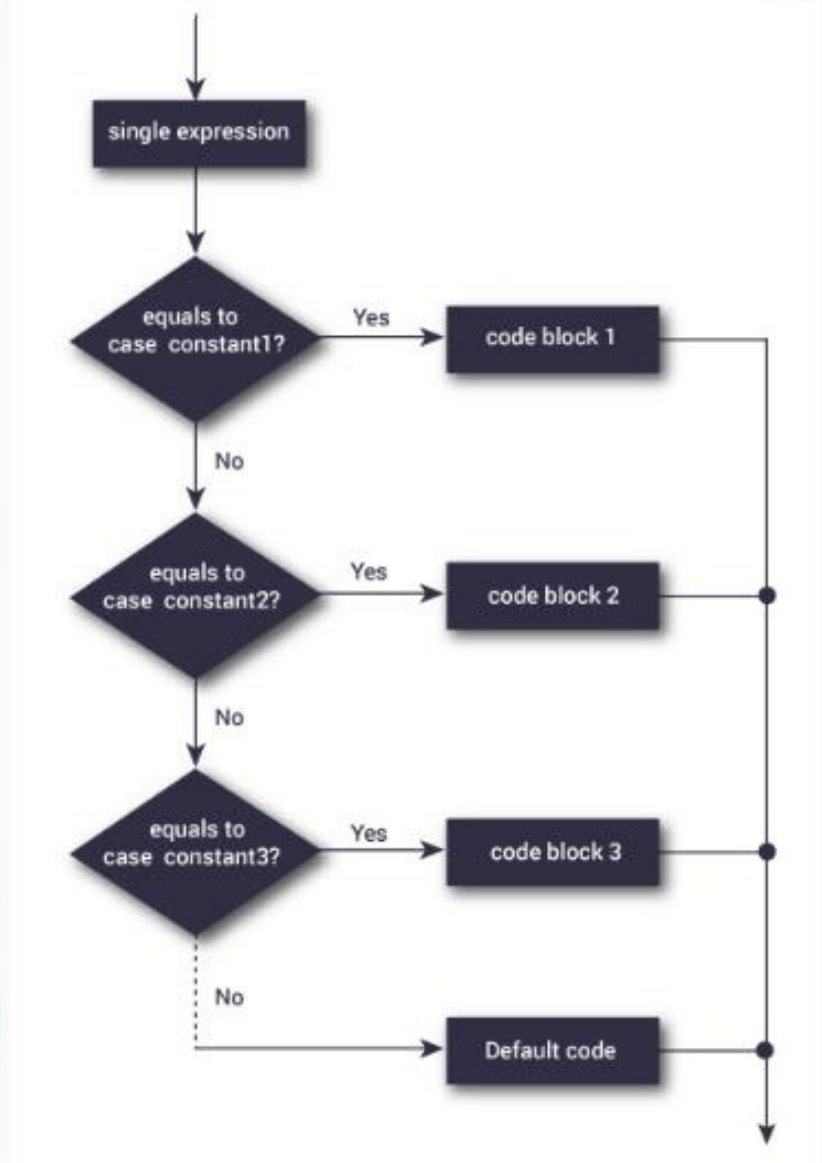
```
switch( expression )
{
    case value-1:
                Block-1;
                Break;
    case value-2:
                Block-2;
                Break;
    ……
    ……
    ……
    case value-n:
                Block-n;
                Break;
     default:
                 Block-1;
                 Break;
}
Statement-x;
```

```
switch(expression)
{
  case constant 1:
            // statements;
               break;
  case constant 2:
             // statements;
                break;
  case constant n:
             // statements;
                break;
  ……
  ……..
  ……
   default:
            // statements;
}
```

Cont…

Flow chart of

Switch case



Program using switch case

Click here to see program of switch case

Like this you can perform any program.

Some Rule for Switch case

- The switch expression must be of integer or character type
- The case value must be integer or character constant
- The case value can be used only inside the switch statement
- The break statement in switch case is not must. It is optional. If there is no break statement found in switch case, all the cases will be executed after matching the case value
- It is known as fall through state of C switch statement
- Do not repeat any cash else it gives an error.

Looping statement

- The loops in C programming is use to execute block of code n numbers of time.
- Basically it is use to execute same task several time.
- In other words, it iterates a code or group of code many times.
- To control the loop execution we have to use condition. It will executer till condition is fulfill else loop entered into the infinite loop.
- In c we have two types of loop entry control and exit control.

C provide three loops

1) For loop (entry control)
2) While loop (entry control)
3) Do while loop (exit control)

For loop in C Programming

Is use to execute block of code till the condition become true. It will terminate the execution when condition become false.

For loop is known as entry control loop because it check condition first before executing the inside statements.

Block of code must enclosed within { } braces.

It execute code “0” or ‘N” times as condition is given.

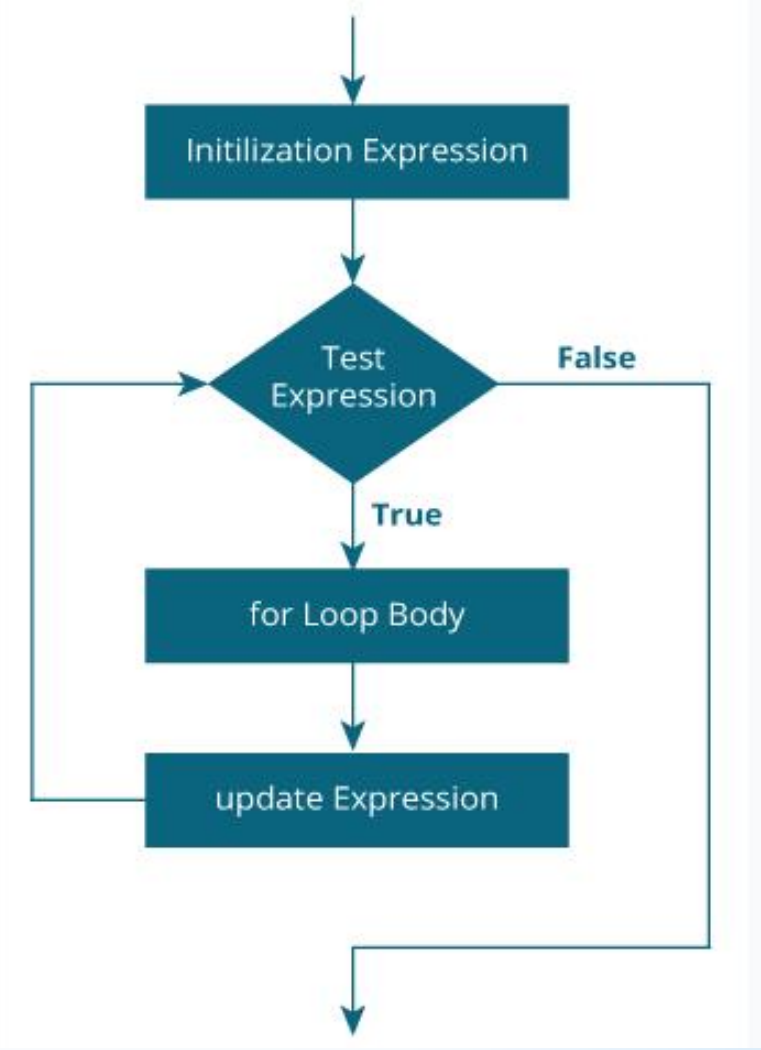
Syntax :

```
for(initialization;condition;incr/decr)
  {
        //code to be executed
  }
```

```c
#include <stdio.h>

int main()
{
  int i;

  for(i=0;i<10;i++)
  {
      printf("%d ",i);
  }

}
```

Flowchart of for loop

- We can also declare nested For loop. (loop within loop).
- It is not compulsory to give Initialization or increment or Decrement statement.
- If you not declare it, it start With 0 and increment with 1.
- If condition is not satisfy than There may be a chance to To become infinite loop.



Program using for loop

```
//program to print multiplication table of entered number
#include<stdio.h>
#include<conio.h>
void main()
{
        int num,i;
        clrscr();
        printf("\n enter number to print table : ");
        scanf("%d",&num);
        printf("\n Table of number %d is ",num);
        for(i=1;i<=10;i++)
        {
                printf("\n %d * %d  = %d",num,i,(num*i));
        }
        getch();
}
```

While loop

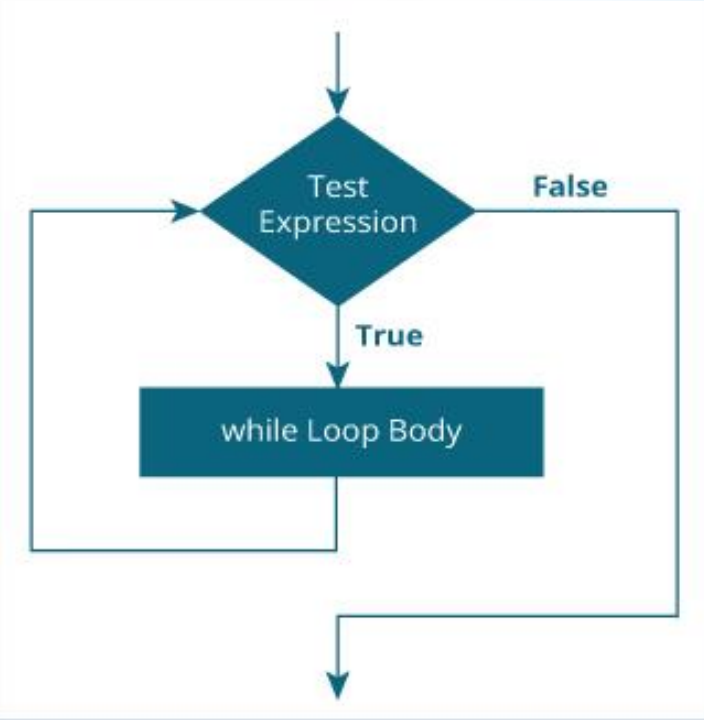
- It iterates the code until condition is false.
- Here, condition is given before the code.
- So code may be executed 0 or more times.
- It is better if number of iteration is not known by the user.
- It is also known as entry control loop. Because condition is check before entering into loop.
- The syntax of while loop

```
while(condition)
  {
        //code to be executed
        //increment or decrement
  }
```

```c
#include <stdio.h>

int main()
{
  int i=3;

  while(i<10)
  {
 printf("%d\n",i);
 i++;
  }

}
```

Flowchart of while loop

- It check condition before entering into loop.
- Not like for loop it have increment or decrement statement inside loop.
- If you don't know stopping condition than it may go in infinite loop.



Program using while loop

```
//Program to print multiplication table using while loop
#include<stdio.h>
#include<conio.h>
void main()
{
        int num,iter=1;          //declaring variable num and iter of type int
        clrscr();                 //function to clear screen
        printf("\n enter number to print multiplication table : ");
        scanf("%d",&num);
        printf("\n multiplication table of entered number is");
        while(iter<=10)
        {
                printf("\n %d * %d = %d", num,iter,(iter*num));
                iter++;    //is same as num = num + 1
        }
        getch();       //function to hold out put on screen
}
```

Do while loop

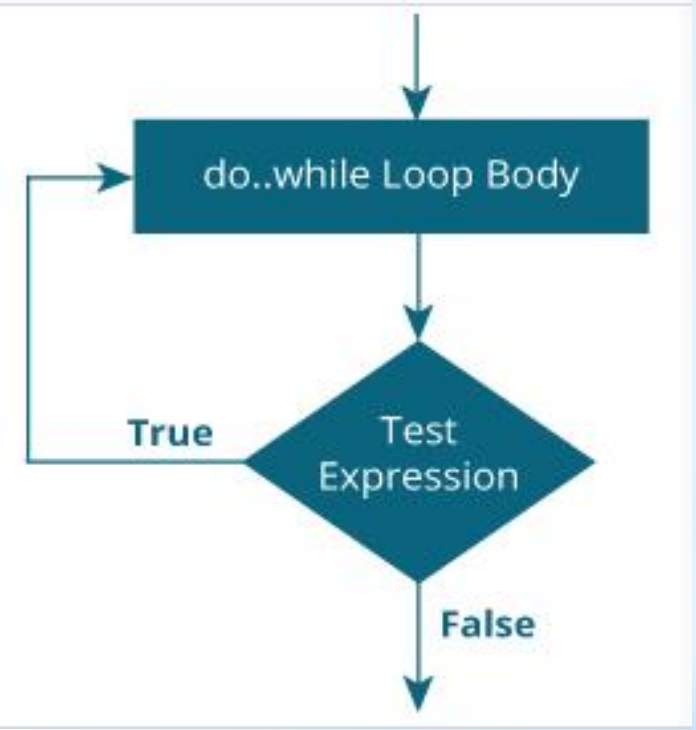
- It iterates the code until condition is false. Here, condition is given after the code.
- So at least once, code is executed whether condition is true or false. It is better if you have to execute the code at least once.
- It is known as exit control loop.
- The syntax of do-while loop in c language is given below:

```
do
{
        //code to be executed
}
while(condition);
```

```c
#include <stdio.h>

int main()
{
  int i=1;

  do
  {
   printf("Value of i is %d\n",i);
  i++;
  }while(i<=4 && i>=2);

}
```

Flowchart

- You can see in the flowchart Condition is check after the code At the time of exit from the loop.
- That's why it going to executed one more time if condition is true or false.



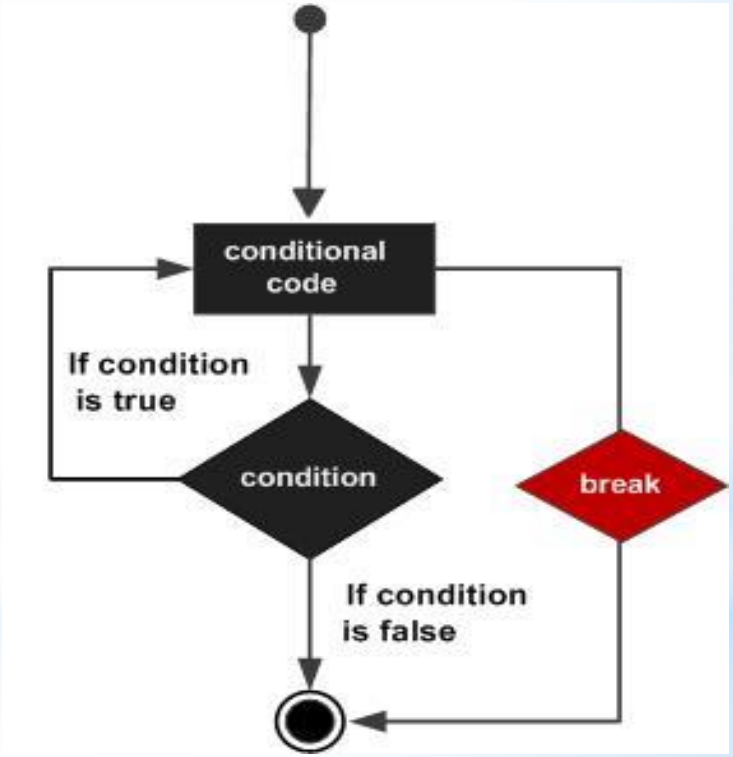
Program using do while loop

```
//Program to print multiplication table using do while loop
#include<stdio.h>
#include<conio.h>
void main()
{
        int iter=1,num;
        clrscr();
        printf("\n enter number to print multiplication table : ");
        scanf("%d",&num);
        do
        {       printf("\n %d * %d = %d", num,iter,(num*iter));
                  iter++;
        }while(iter<=10);
        getch();
}
```

Break, continue, goto and exit statement

Break statement : is use to break the execution based on given condition.

- If condition id fulfill than it come out from the execution of loop or any other code.
- For that **break** keyword is use.

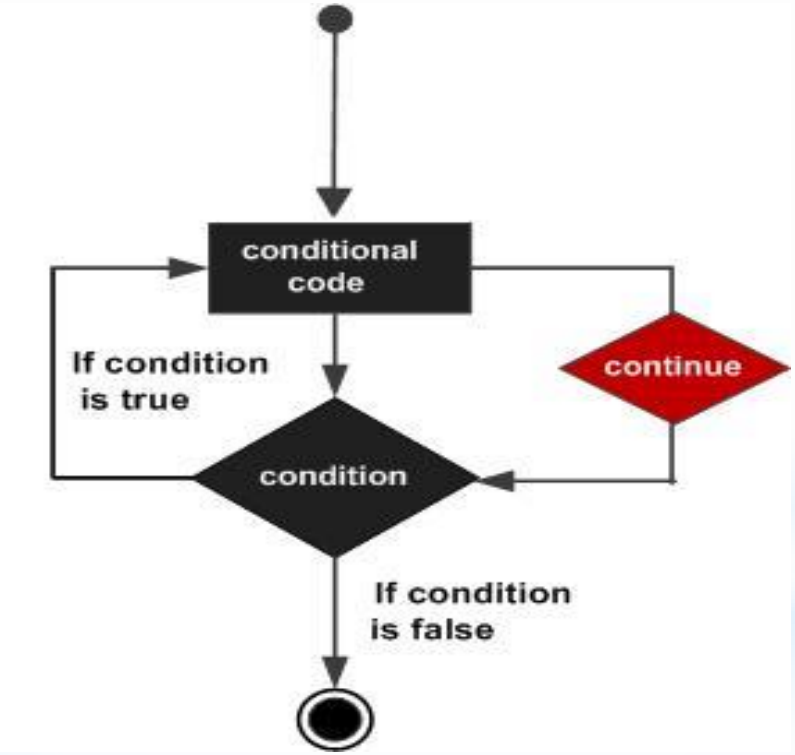


Program using break statement

```
#include <stdio.h>
#include <conio.h>
void main()
{
        int i=1;
        clrscr();
        for(i=1; i<=10; i++)
        {
                printf("%d \n",i);
                if(i==5)
                {
                        break;
                }
        }
        getch();
}
```

Continue : it is use to continue the iteration or execution of program based on given condition

- For that **continue** keyword is use.



Program using continue statement

```
#include <stdio.h>
#include <conio.h>
        void main ()
        {
                int a = 10;
                 do
                {
                     if( a == 15)
                     {
                                a = a + 1;
                                continue;
                      }
                      printf("value of a: %d\n", a);
                      a++;
                      } while( a < 20 );
                      getch();
        }
```

Goto statement : It is use to jump from one instruction to other in the program.

The goto statement allows us to transfer control of the program to the specified label.

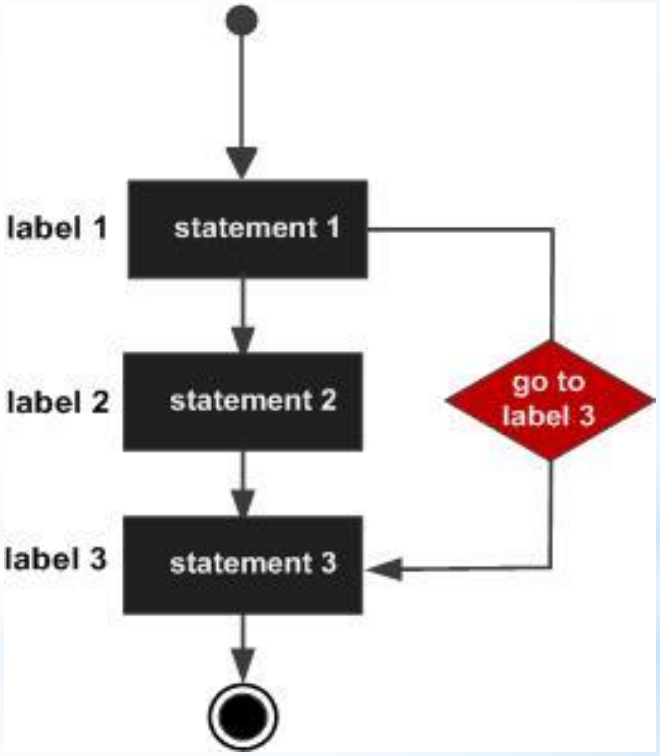
Syntax:

Forward jump

goto label;

.. .

label: statement;



Program using goto statement

```
void main()
{    //age1,age2,age3 is label for goto
            int age;
            clrscr();
            printf("\n enter your age : ");
            scanf("%d",&age);

            if(age<=0)
                        goto age1;
            else if(age>0 && age<18)
                        goto age2;
            else if(age>18 && age<50)
                        goto age3;
            else
                        printf("\n you are allow for voting online not offline");
                        goto last;
            age1:
                        printf("\n please enter valid age");
                        goto last;
            age2:
                        printf("\n you are not allow for voating");
                        goto last;
            age3:
                        printf("\n you are allow for voating at center");
            last:
                        printf("\n bye");
            getch();
}
```

Exit statement

- exit() is a standard library function, which terminates program execution when it is called.
- syntax for a exit statement

  void exit(int status)

- status -> The status in an integer value returned to the parent process.
- Here 0 usually means program completed successfully, and nonzero values are used as error codes. e.g exit(0);
- There are also predefined macros EXIT_SUCCESS and EXIT_FAILURE, e.g. exit(EXIT_SUCCESS);
- In the C Language, the required header for the
- exit( ) function is stdlib.h.

Program using exit

```
#include<stdio.h>
#include <conio.h>
void main ()
{
        printf("Start of the program....\n");
        printf("Exiting the program....\n");
        exit(0);
        printf("End of the program....\n");
        getch();
}
```

Difference between `break` and `exit()`

break	exit()
`break` is a keyword in C.	`exit()` is a standard library function.
`break` causes an immediate exit from the `switch` or loop (`for`, `while` or `do`).	`exit()` terminates program execution when it is called.
`break` is a reserved word in C; therefore it can't be used as a variable name.	`exit()` can be used as a variable name.
No header files needs to be included in order to use `break` statement in a C program.	`stdlib.h` needs to be included in order to use `exit()`.
`break` transfers the control to the statement follows the `switch` or loop (`for`, `while` or `do`) in which `break` is executed.	`exit()` returns the control to the operating system or another program that uses this one as a sub-process.
Example of `break`	Example of `exit()`

```
// some code here before while loop
while(true)
{
    ...
    if(condition)
      break;
}
// some code here after while loop
```

```
// some code here before while loop
while(true)
{
    ...
    if(condition)
      exit(-1);
}
// some code here after while loop
```

Conditional Ternary operator (? :)

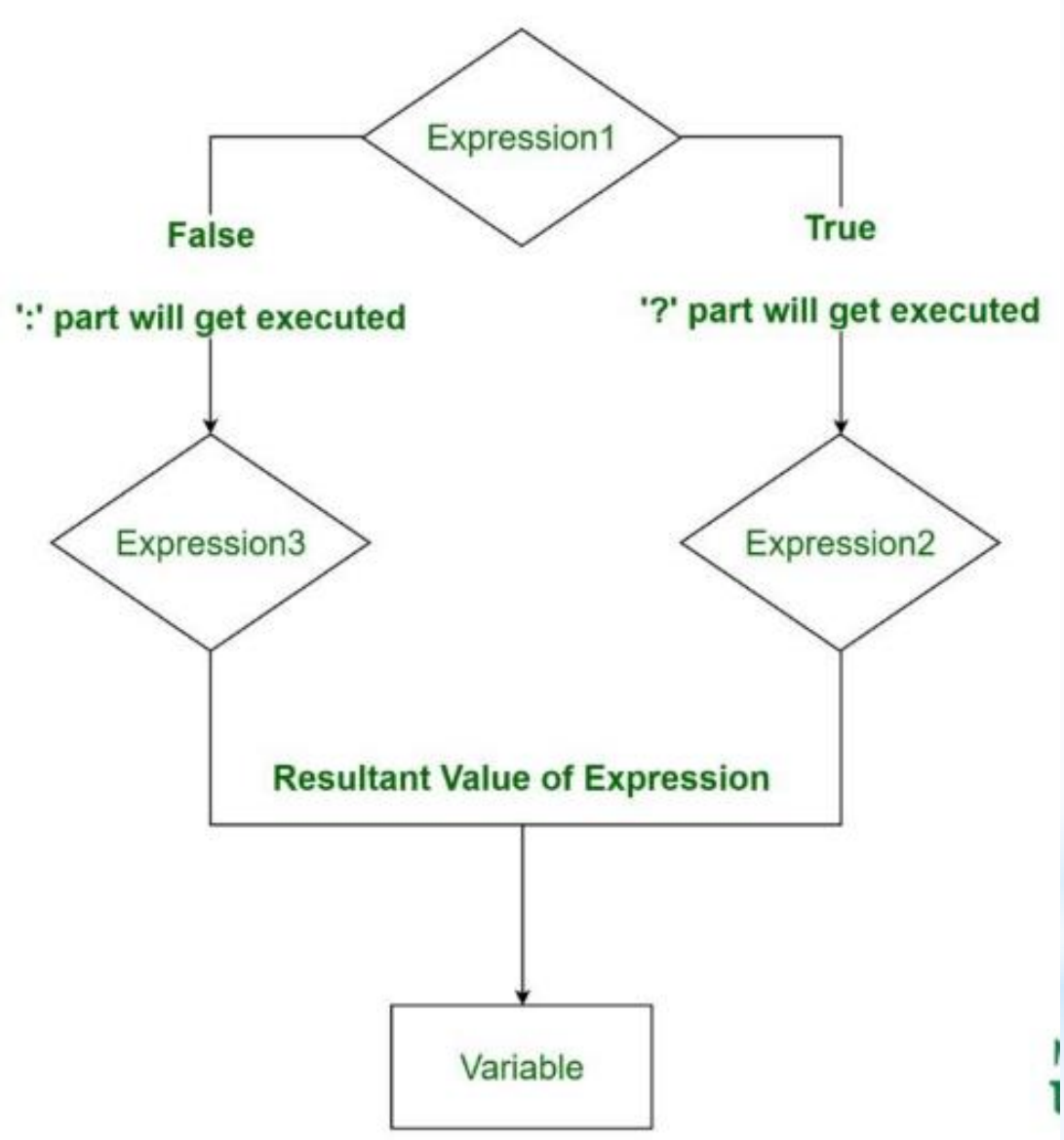
- Conditional operator is similar as if else statement.
- It takes less space rather than if else program.
- It helps to write if else statement in shortest manner.

Syntax : Expression1 ? Expression2 : Expression3

Is similar to if else like

```
if(Expression1)
{
        variable = Expression2;
}
else
{
        variable = Expression3;
}
```

Flowchart of Conditional Ternary operator



Program of Conditional Ternary operator

```
//Program using conditional operator (specical operator ? :)
#include<conio.h>
#include<stdio.h>
void main()
{
        int num;
        clrscr();
        printf("\n enter number to check : ");
        scanf("%d",&num);
        ((num>0)?printf("\n positive"):printf("\n negative"));
        getch();
}
```

Perform below list of programs using all loop

- Write a C program to print all natural numbers from 1 to n. - using while loop
- Write a C program to print all natural numbers in reverse (from n to 1). - using while loop
- Write a C program to print all alphabets from a to z. - using while loop
- Write a C program to print all even numbers between 1 to 100. - using while loop
- Write a C program to print all odd number between 1 to 100.
- Write a C program to print sum of all even numbers between 1 to n.
- Write a C program to print sum of all odd numbers between 1 to n.
- Write a C program to print table of any number.
- Write a C program to enter any number and calculate sum of all natural numbers between 1 to n.
- Write a C program to enter any number and find its first and last digit.
- Write a C program to enter any number and calculate sum of its digits.

- Write a C program to enter any number and calculate product of its digits.
- Write a C program to swap first and last digits of any number.
- Write a C program to enter any number and print its reverse.
- Write a C program to enter any number and check whether the number is palindrome or not.
- Write a C program to enter any number and check whether it is Prime number or not.
- Write a C program to enter any number and check whether it is Armstrong number or not.
- Write a C program to enter any number and check whether it is Perfect number or not.
- Write a C program to enter any number and check whether it is Strong number or not.
- Write a C program to print all Prime numbers between 1 to n.

- Write a C program to print all Armstrong numbers between 1 to n.
- Write a C program to print all Perfect numbers between 1 to n.
- Write a C program to print all Strong numbers between 1 to n.
- Write a C program to enter any number and print its prime factors.
- Write a C program to find sum of all prime numbers between 1 to n.
- Write a C program to print Fibonacci series up to n terms.

Reference for more study

https://www.tutorialspoint.com/cprogramming/c_loops.htm

https://www.programiz.com/c-programming

https://www.guru99.com/c-if-else-statement.html

https://www.geeksforgeeks.org/conditional-or-ternary-operator-in-c-c/

https://www.guru99.com/c-storage-classes.html

https://www.javatpoint.com/storage-classes-in-c

https://www.guru99.com/c-loop-statement.html

https://beginnersbook.com/2014/01/c-loops-examples/