04/10/24

(Unit - 2)

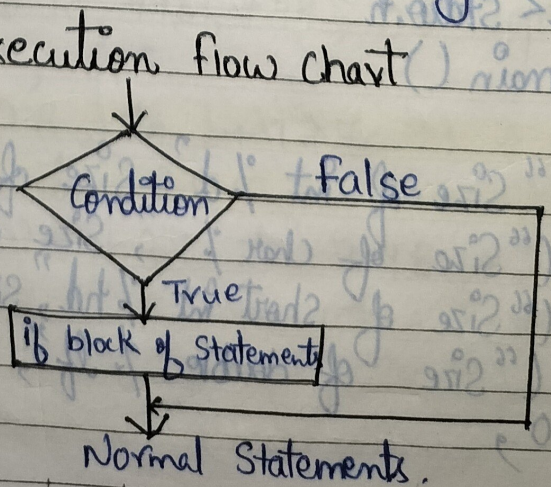
1. **Branching Statement**
   ↳ If, If - Else, If - Else - If Ladder, Nesting of If

2. **Looping Statement**
   ↳ for, while, do - while.

3. **Jump Statement**
   ↳ Break, return, Continue, goto

1. **Branching / Decision Making Statements** : The statements in the program that helps to transfer the control from one part to other parts of the program. Facilitates program in determining the flow of Control.

↳ If Statement



```
#include <stdio.h>
#include <conio.h>
void main ()
{
int age ;
clrscr ();
printf ("Enter the age:");
Scanf ("%d", & age);
if (age >= 18)
printf ("\n The person is eligible to vote:");
getch ();
}
```

```
if (1) {
---------
---------printf ("Eligible for voting")
}
```

anything i.e., non zero (TRUE), it will be printed as it is otherwise zero (FALSE).

* Evalution.

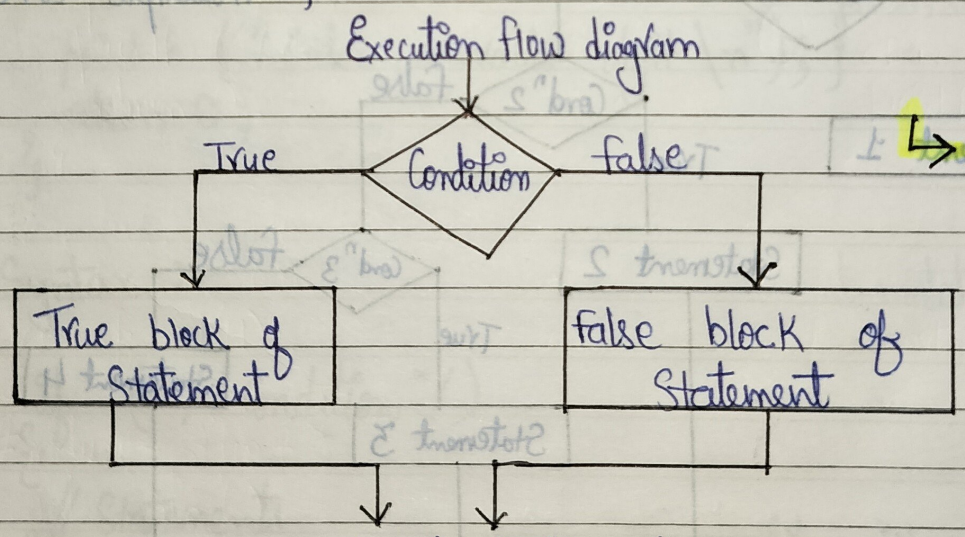
OUTPUT

Enter the age : 30

The person is eligible to vote.

Syntax : if (condition) {
Statement ;
}

2.



↳ if - Else Statement

```
#include ---
---------
{ ---------
---------
---------
* else
printf ("\n ---- not ---)
---------
{
```

OUTPUT

Enter the age : 11

The person is not Eligible to vote.

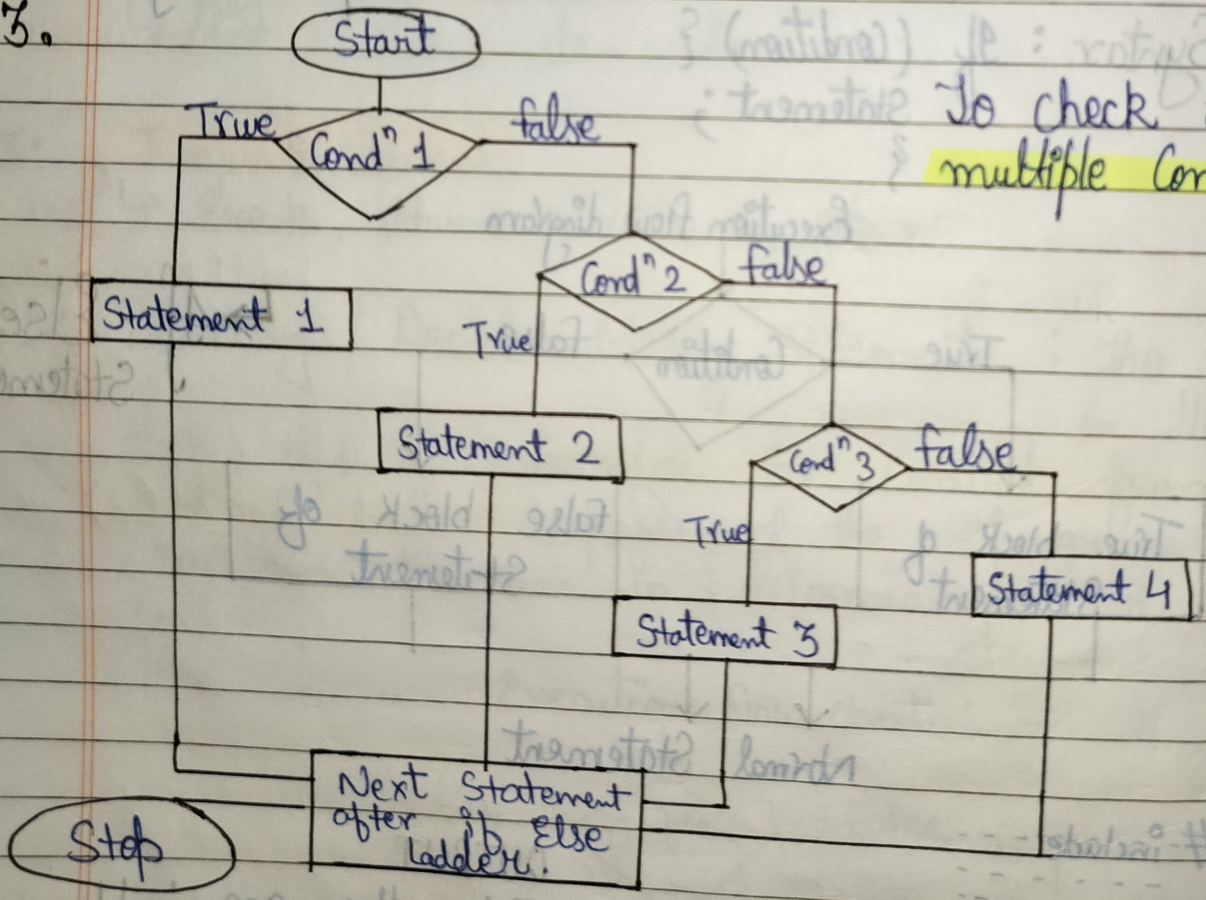
Syntax : If (Condition)
{
- - - - -
True block of Statements ;
- - - - -
}
else
{
- - - - -
false block of Statements ;
- - - - -
}

If Else Statement takes care of both true & false Conditions.

If block is for True Condition.

Else block is for false Condition.

3.



To check the multiple Conditions.

↳ If - Else - If Statement

Ex:-
```
#include <stdio.h>
int main()
{ int weekday;
printf("Enter any digit (1-7) assuming 1 for Monday and 1 for Sunday: ");
scanf("%d", &weekday);
if (weekday == 1) {
printf("Today is Monday\n");
} else if (weekday == 2) {
printf("Today is Tuesday\n");
} else if
    :
    :
} else if (weekday == 7) {
  printf("Today is Sunday\n");
} else {
printf("Enter a valid digit\n"); }
  return 0;
}
```

Syntax :-

```
if (/* Condition */)
{
 // Statements
}
else if (/* Condition */)
{
//Statements
}
  :
  :
```

4. Nesting of if : writing entire if-else statement within entire body of another if statement or the body of an else statement. This is called 'nesting' of if.

Syntax :-

```
if (/* Condition */)
{
 if (/* Condition */)
{
  // statements
}
else
{
 // Statements
}
}
else
{
// statements
}
```

Dangling Else problem → due to prbm in writing syntax

The dangling else prbm is syntactic ambiguity. It occurs when we use nested if, due to scarcity of else block. When there are multiple 'if' statements, the 'else' part doesn't get a clear view with which "if" it should combine.

```
if (Test ExpA)
    if (Test ExpB)
else
    StmAf
```

OR

```
if (Condition) {
}
if (Condition 1){
}
if (Condition 2){
} else
{ }
```

Here, we want to pair the outermost if with the else part. But the else part doesn't get a clear view with which "if" Cond" it should pair. This leads to inappropriate results in programming (due to wrong interpretation by Compiler.

Solutions

Use of null else.
Use of braces to enclose the true action of the second if.

with null else

```
if (Test Exp A)
   if (Test Exp B)
      Stm Bf ;
   else  <-- Empty
      ;
else
   Stm Af ;
```

with braces

```
if (Test Exp A)
{
   if (Test Exp B)
      Stm Bf ;
}
else
   Stm Af ;
```

WAP to find largest among 3 integers.

```
#include <stdio.h>
int main ()
{ int c = 10, b = 22, a = 9;
  if (a >= b)
  { if (a >= c)
      printf ("%d is the largest no", a);
    else
      printf ("%d is the largest no", c);
  }
  else
  { if (b >= c)
      printf ("%d is the largest no", b);
    else
      printf ("%d is the largest no", c);
  }
}
```

- if-else using logical AND operator.

```
#include <stdio.h>
int main()
{ int a=11, b=2, c=9;
  if (a>=b && a>=c)
    printf ("%d is the largest", a);
  else
    if (b>=a && b>=c)
      printf ("%d is the largest", b);
    else
      printf ("%d is the largest", c);
  return 0;
}
```

WAP to check +ve, -ve or 0 using if else.

```
#include <stdio.h>
int main ()
{
 int num;
printf ("Enter a number:");
scanf ("%d", &num);
if (num >0)
 printf ("Number is POSITIVE");
else if (num <0)
printf ("Number is NEGATIVE");
else
 printf ("Number is ZERO");
}
```

WAP to check alphabet

```
#include <stdio.h>
int main()
{ char ch;
 printf ("Enter any character:");
 scanf ("%c", &ch);
 if ((ch>='a' && ch<='z') ||
 (ch>='A' && ch<='Z'))
                  check ASCII value
 { printf ("Character is Alphabet");
 }
 else
 { printf ("character isn't an alphabet");
 }
 return 0;
}
```

NOTE: ASCII Value: a = 97, A = 65, z = 122, Z = 90

WAP to check divisibility of a number.

```
#include <stdio.h>
int main ()
{ int num;
printf ("Enter a number");
Scanf ("%d", & num);
if ((num %5 == 0) && (num %11 == 0))
{ printf ("Num is divisible by 5 & 11");
  } else
{ printf ("Num is not divisible by 5 & 11");
}
  return 0;
}
```

```
if (!(num %5) && !(num %11)
printf ("Num is divisible by
  5 & 11");
else
printf ("Num is not divisible
  by 5 & 11");
```

Ex- Num = 54

54%5 && 54%11

!(4) && !(11)

!(T) && !(T)

F && F

→ Not Executed

num = 55

55%5 && 55%

!(0) && !(0)

!(f) && !(f)

T && T

→ Executed

WAP to enter marks & find % and grade.

```
#include <stdio.h>
  int main()
{ float chem, bio, math, Eng, phy., per;
printf ("Enter the marks of Chemistry, biology, mathematics,
  English and physics");
Scanf ("%f %f %f %f %f", & Chem, & bio, & math, & Eng,
  & phy);
per = ( chem + bio + math + Eng + phy) / 5.0 ;
if (per >= 90) {
  printf ("Grade A"); }
else if (per >= 80) {
  printf ("Grade B"); }
else if (per >= 70) {
  printf ("Grade C"); }
```

```
else if (per >= 60) {
    printf ("Grade D"); }
else if (per >= 40) {
    printf ("Grade E"); }
else { printf ("Grade f");
  }
  return 0;
}
```

WAP to Calculate Profit.

```
#include <stdio.h>
  int main ()
{
 int CP, SP, amt;
 printf ("Enter the cost price of an item :");
  Scanf ("%d", & CP);
printf ("Enter the Selling price of an item:");
  Scanf ("%d", & SP);
 if( SP> CP)
{
 amt = SP- CP
 printf ("Profit is %d", amt);
}
else if (CP> SP)
{
 amt = CP- SP
  printf ("loss is %d", amt); }
else { printf ("Neither profit nor loss");
// when CP = SP }
return 0;
}
```

Body loop.

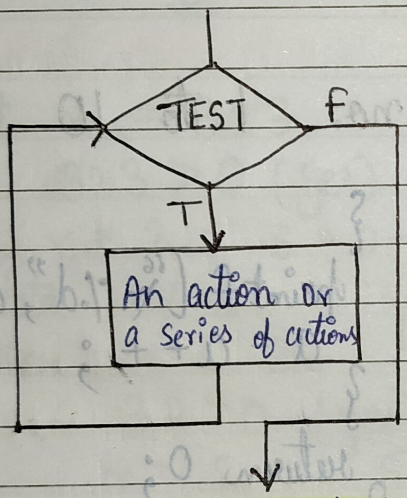
Page No.:
Date: 09/10/24

Q. Loops :— Loops are divided into 2 parts.

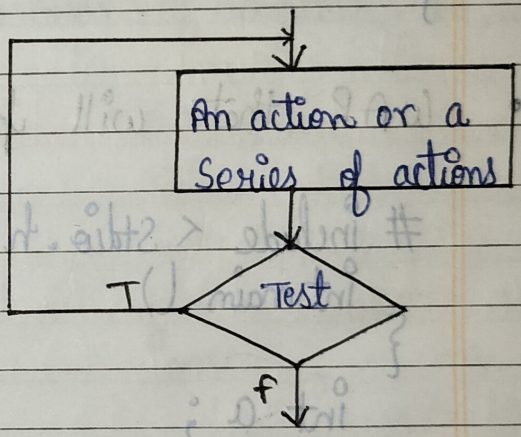
1. Pre-Test (Entry Control loop) <-- while, for
2. Post-Test (Exit Control loop) --- do-while

- Pre - first check Condition then Execute & Post execute first then Check Condition.

Loop :- A loop allows one to execute a statement or block of statements repeatedly



Pre test loop.



Post test loop.

While Statement is a pre test loop. The Basic Syntax of the while statement as follows :

```
while (Test Expe.)
{
Stm T
}
```

No Semi Colon

Body of loop.

No Semi Colon

- Basically, Loop require
  - ↳ Initialization (assigning the initial value).
  - ↳ Test Expression/Condition
  - ↳ Updation

- ```
  #include <stdio.h>
  int main ()
  {
  Int c;
  c = 5; // Initialization
  while (c>0)
  { // Test Expression
  printf ("\n%d", c);
  c = c-1; // updating
  }
  return 0;
  }
  ```

This loop contains all the parts of a while loop.

When Executed in a program this loop will output:

5
4
3
2
1

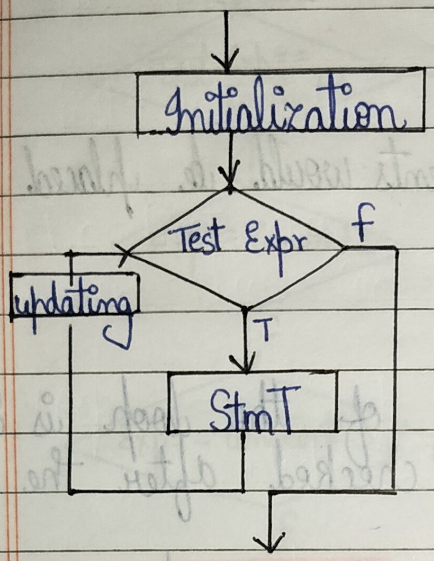
- WAP that will print a no. 1 to 10.

```
#include <stdio.h>
int main ()
{
int a;
a = 1
while (a < 11)
{
printf ("\n%d", a);
a = a++;
}
return 0;
}
```

- WAP that will print first n even numbers.
```

Page No.:
Date: 10/10/24

For statement is a pre test loop. The flow chart of 'For' looping is as follows.



WAP to print the Sum of digits of any no.

```
int main () {
int n, S = 0, r;
printf ("\n Enter the No");
Scanf ("%d", &n);
for (; n > 0; n /= 10) {
    r = n % 10 ; S = S+r; }
printf ("\n Sum of digits %d", S);
```

↳ 31 % 10
r = 1

↳ 31/10
q = 3

↳ n /= 10
n = n/10;

Let n = 323
323 > 0 (yes)
r = 3 ; S = 0+3 = 3
update n /= 10 = 323/10 = 32.
then n = 32
32 > 0 (yes)
r = 2 ; S = 3+2 = 5.
n /= 10 = 32/10 = 3

* Initialization → Condition → statement to be Executed
n /= 10 will update value of n
if n = 12345 > 0 (yes)
r = 5 ; S = 5.
updation n /= 10
= 4.

* Runs as long as a Cond" TRUE
↳ WHILE LOOP (used when no. of iterations is known.)

Print "Hello world" 5 times using 'for' loop.

```
#include <Stdio.h>
int main ()
{
  for (int i = 0 ; i < 5 ; i++)
{
  printf ( " Hello world ! \n") ;
}
return 0 ;
}
```

Do-while statement is a Post test loop. Basic structure are as follows:

```
do
{
  statement ; /* body of statements would be placed here */
}
while (Test Expression);
```

↳ Do-while statement, the body of the loop is Executed first & the test Exp. is checked after the loop body is Executed.
It runs at least once & then Continues if a Condition is true.

1 tab = 8 char.

```
int i;
for(i=1; i<=5; i++)
{
printf("%d\n", i);
}
```

↳ O/P: 1 2 3 4 5

without it execu upto def from the con is true

out of Body of fn if Cond ↳ outside


```
#include <stdio.h>
int main()
{
  int row, col;
  for (row=1; row<=4; ++row)
  {
    for (col=1; col<=row; ++col)
    printf("%d\t", row);
    printf("\n");
  }
return 0;
}
```

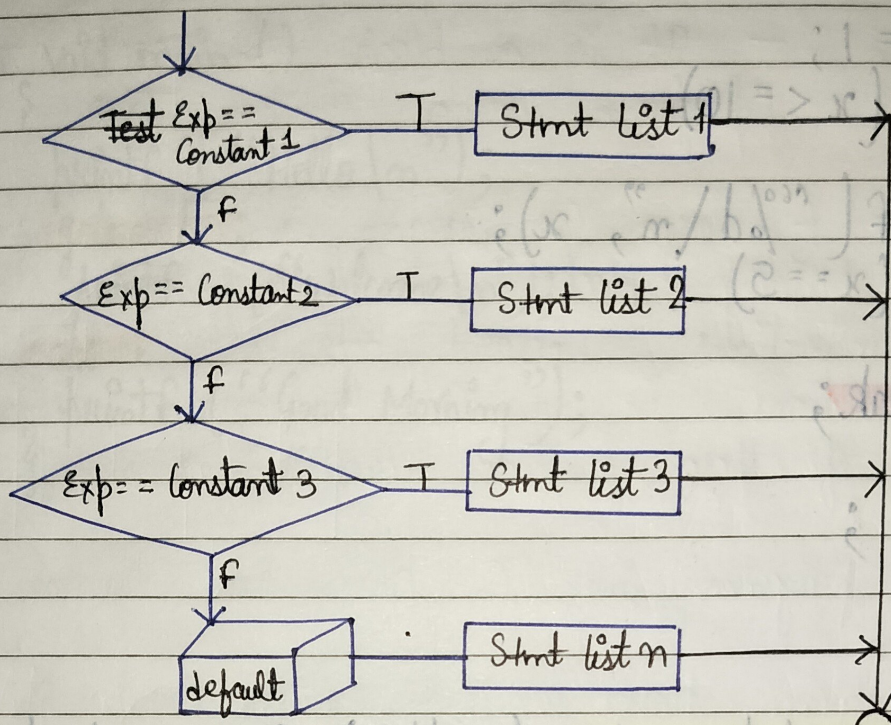
O/P:
1
2 2
3 3 3
4 4 4 4

```
*
* *
* * *
* * * *
* * * * *
```

(Reverse Loop: initialization - max value)

Page No.:
Date: 11/10/24

The Switch Statement



The C Switch Construct.

```
Switch (Expr)
{
Case Constant 1 : Stmt list 1;
break;
Case Constant 2: Stmt list 2;
break;
Case Constant 3: Stmt list 3;
break;
- - - - - - - - -
default : Stmt lisn ;
}
```

"Break" Keyword is used to take the Control of the Construct.

without this it executes upto default from where the condition is true.

out of the Body of Switch, if Cond" flase ↳ outside the loop.

↳ (optional) ⇒ Invalidcode.

Fixed no. of choices
↳ (vowels & consonent, weekdays).

Inequalities (>, <) → N/A

The Switch differs from the else-if (less complexity) in that Can test only for equality, whereas if Conditional Expression Can be of a test expression involving any type of relational operators and/or Logical operators.

```
void main ()
{
 int x = 1;
 while (x <= 10)
 {
  printf ("%d\n", x);
  if (x == 5)
  {
   break;
  }
  x++;
 }
}
```

Continue statement: (skipping) It is used to move the control to the next repetition of the loop. We can use this **only inside the loops.**

```
void main ()
{
 int x = 1;
 while (x <= 10)
 {
  if (x % 3 == 0)
  {
   x++;
   continue;
  }
  printf ("%d\n", x);
  x++;
 }
}
```

O/P :-
1
2
4
5
7
8
10

Go To Statement :

```
void main ()
{
printf ("Hello\n");
goto abc;
printf ("Welcome\n");
abc;
printf ("Good Morning");
}
```