~~Managing I/P & O/P Operations~~

## Decision making & Branching:

(2way decision making)

→ A Decision making structures requires that the programmer specifies one or more conditions to be evaluated or tested by a pgm.

→ Decision making is abt deciding the order of execution of statements based on certain conditions or repeat a group of statements until certain specified conditions are met.

Decision making capabilities by supporting following statements:

1. if statement
2. Switch statement
3. goto statement

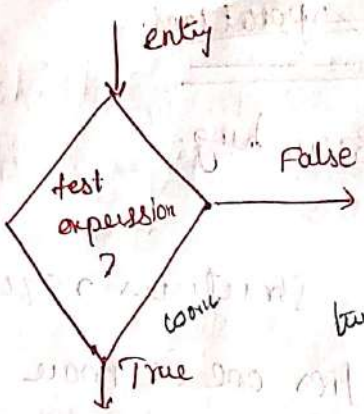
## Decision making with if Statement

→ The if statement is a powerful decision making statement and is used to control the flow of execution of statement.

→ It is basically a two-way decision statement and is used in conjunction with an expression.

→ It takes following form: ~~if else~~

if (test expression)




4 types:

1. Simple if statement
2. If..else statement
3. Nested if..else statement
4. else if ladder.

## ① Simple if Statement:

A general form of Simple if statement:

```
if (test expression)
{ statement-block ;
}
statement x;
```

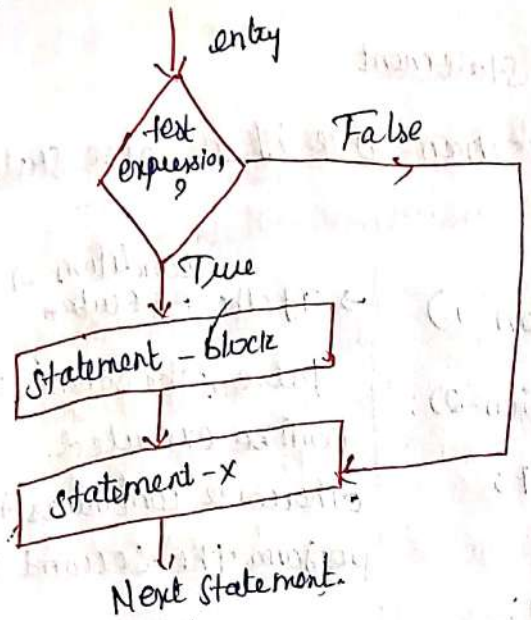
if (n>0)
state - true - display
if not - true @ then jump.

Statement-block → a single statement or a groups of statem

→ If the test expression is true, the statement-bloc will be executed.

→ otherwise the statement-block will be skipped and execution will jump to the statement x.

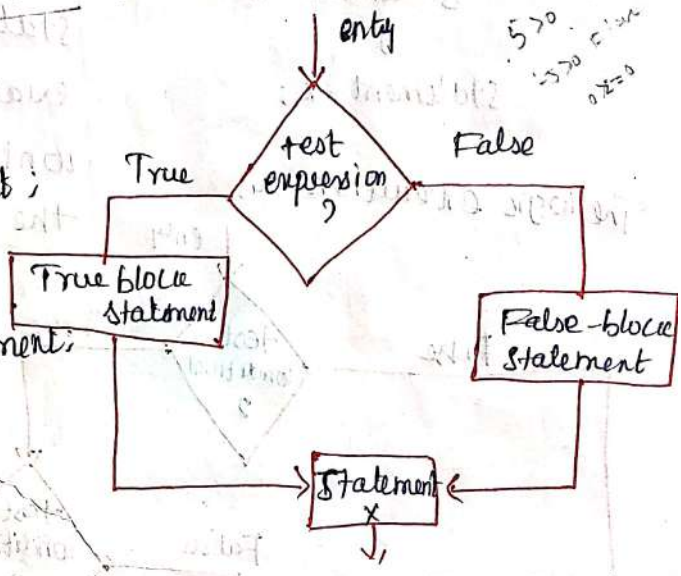
→ ~~Remember~~ Remember, when the condition is true, both the statement-block and statement x-are executed in sequence.




## If - else Statement:

→ It is an extension of the simple if statement.

```
if (test expression)
{ true-block statement;
}
Else
{ false- block statement;
}
Statement-x;
```




→ If the test expression is true, then the true-block statement(s) immediately following the if statement are executed, otherwise, the false-block statements are executed.

→ In either case, either True-block (or) False-block will be executed, not both.

## Nesting of if-else statement.

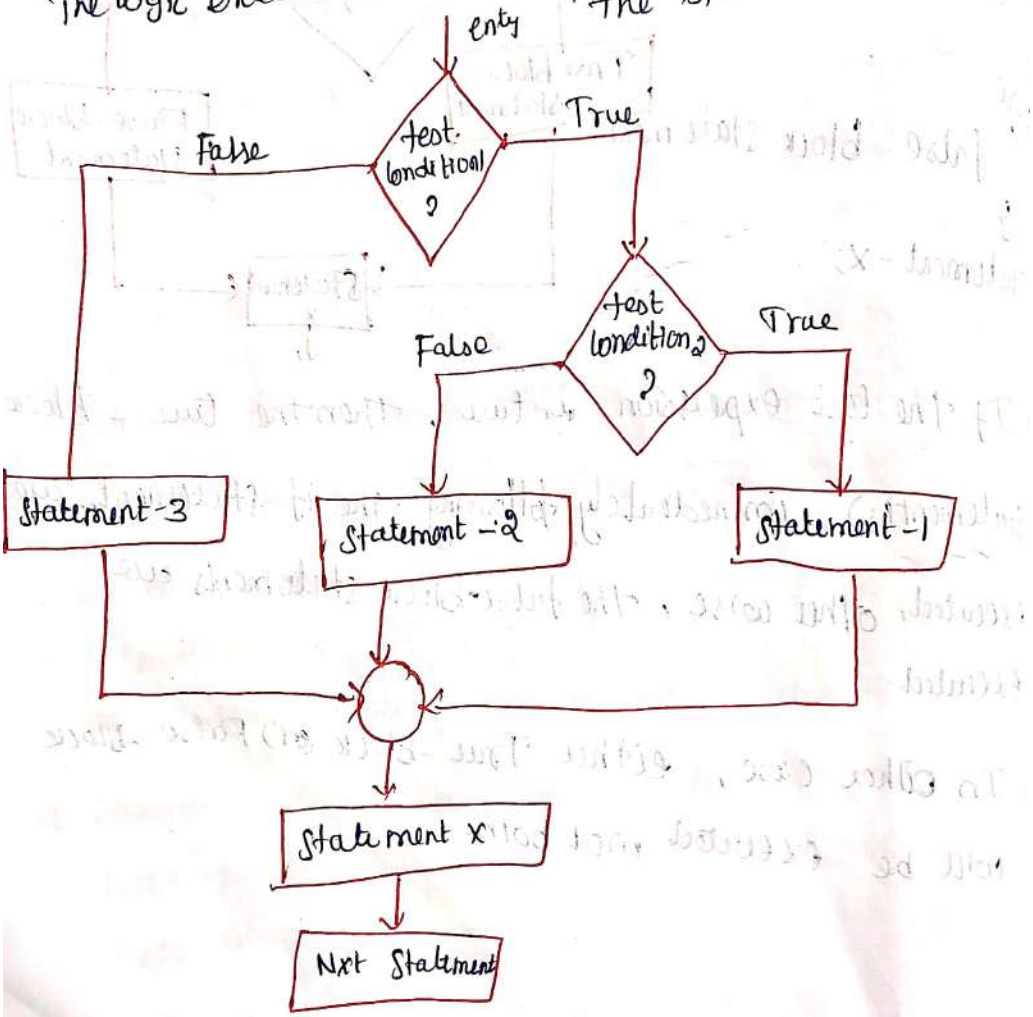
→ use more than one if.... else statement in nested form.

```
if (test condition-1)
  { if test condition-2);
    { statement-1;
    }
  else
    {statement 2;
    }
  }
else
  { statement -3;
  }
statement -x;
```

→ If the condition -1 is ~~execution~~ false, the statement-3 will be executed. otherwise continues to perform the second test.

→ If condition -2 is true the statement 1-will be evaluated. otherwise statement -2 will be evaluated and then the control is transferred to the statement-X.

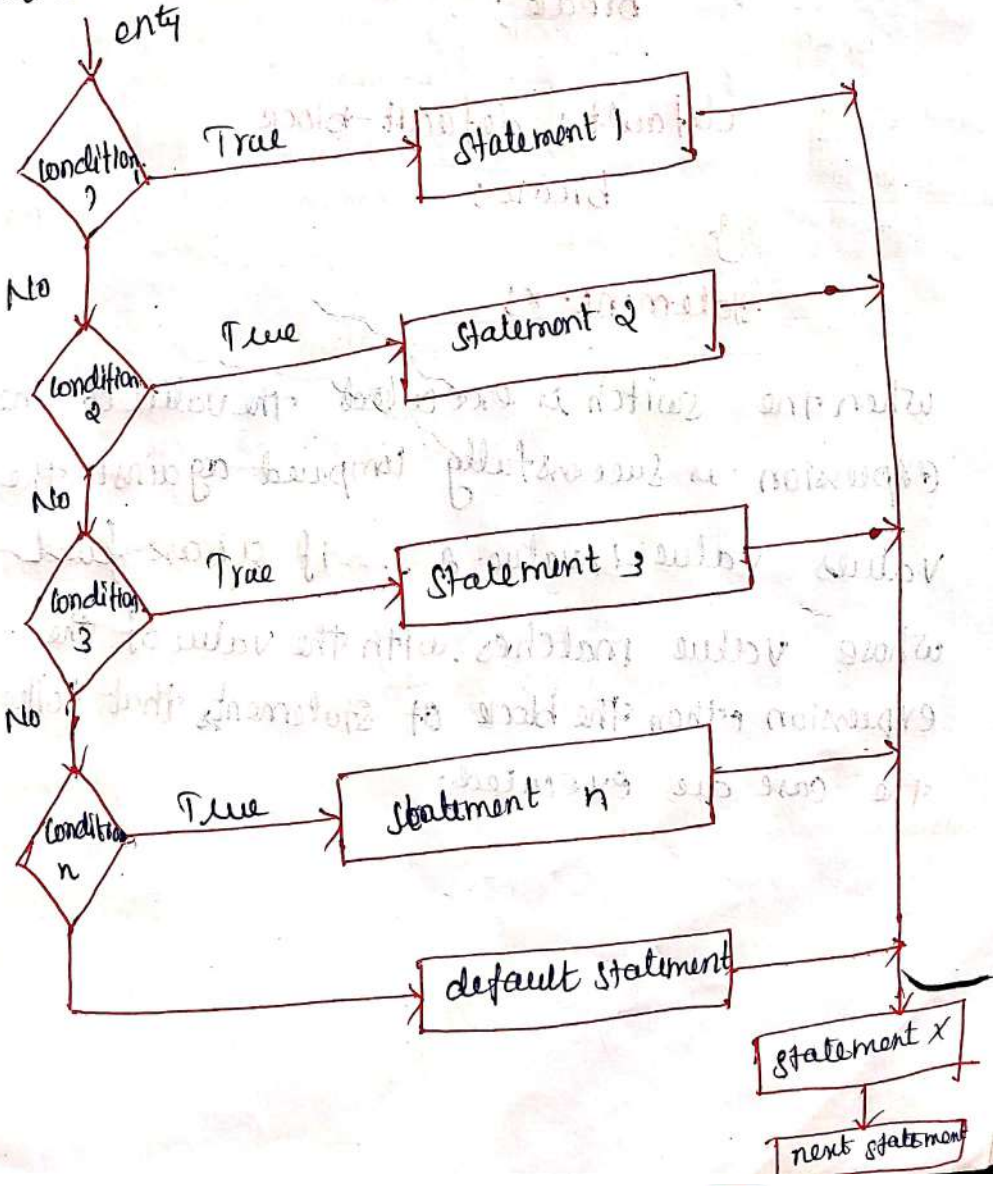
The logic execution: as:



## The else if Ladder:

→ Multipath decisions are involved. It is a chain of ifs in which the statement associated with each else is an if.

```
if (condition 1)
   statement-1;
else if (condition 2)
   statement-2;
else if (condition 3)
   statement-3;
else if (condition n) statement n;
else default-statement ;
statement-x;
```



## ② Switch Statement:

→ Multi way decision statement, known as switch.

→ The Switch statement tests the value of a given variable or expression against a list of case values and when a match is found a block of statements associated with that case is executed.

```
switch (expression)
{
  case value-1 : block-1
    break;
  case value-2 : block-2
    break;
  ....
  default : default-block;
    break;
}
Statement-x;
```

When the switch is executed the value of the expression is successfully compared against the values value-1, value-2 .... if a case found whose value matches with the value of the expression then the block of statements that follows the case are executed.

Sample pgm.

```
#include <stdio.h>
#include <conio.h>
void main()
{
int day;
clrscr();
printf("Enter a day : ");
scanf("%d",&day);
if(day<=7)
{switch(day)
{case 1: printf("Sunday");
         break;
 case 2: printf("Monday");
         break;
 case 3: printf("Tuesday");
         break;
 case 4: printf("Wednesday");
         break;
 case 5: printf("Thursday");
         break;
 case 6: printf("Friday");
         break;
 case 7: printf("Saturday");
         break;
default: printf("This case values does not found");
         break;
}
}
getch();
}
```

O/p: enter a value : 5

Thursday.

( -3 - -3. ?
The value does not found)

Q -

Key Rules:

→ It must be an integral type.
→ case labels must be constants or constant expression.
→ The case labels must be unique.
→ " " end with semi colon.
→ The break statement transfers the control out of the switch statement.
→ The break statement is optional, That is two or more case labels may belong to the same statement.
→ The default label: If present, it will be executed when the expression does not find a matching case labels.
→ There can be at most one default label.
→ The default may be placed anywhere but usually placed at the end.
→ It is permitted to nest switch statements.