

INDRODUCTION



1 SEQUENCE CONTROL B/W STATEMENTS:

Sequence Control b/w Statements:

are **One-in ,One-Out** control statement it means that each statement there is only **one entry point one exit point**. Each One-in, One-out control statement may include internal branching, looping but one exit point.

Forms of Statement-Level Sequence Control

Three main forms are explained below:

1. Composition

Statement may be placed in a textual sequence so that they are executed in a order.

2. Alternation

Two Sequence of statement may be form alternatives so that one or the other sequence is executed but nit the both can be executed

3.Iteration:

A sequence of statements may be executed a number of times until if satisfied a particular condition

Sequence Control b/w Statements:

As we know statement are of two types:

1. Basic Statement:

These statement are written in a single line or also known as the **statement level sequence control**

Basic statement include the following:

- i. Assignment Statement
- ii. Input Statement
- iii. Output Statement
- iv. Go to Statement
- v. Break Statement
- vi. Continue Statement

Explicit Sequence Control

The sequence control b/w basic statement will be straightforward means top to down as describing further:

Sequence Control b/w Statements:

- **Start**
- **input statement**
- **processing statement**
- **output statement**
- **End**



Go to Statement:

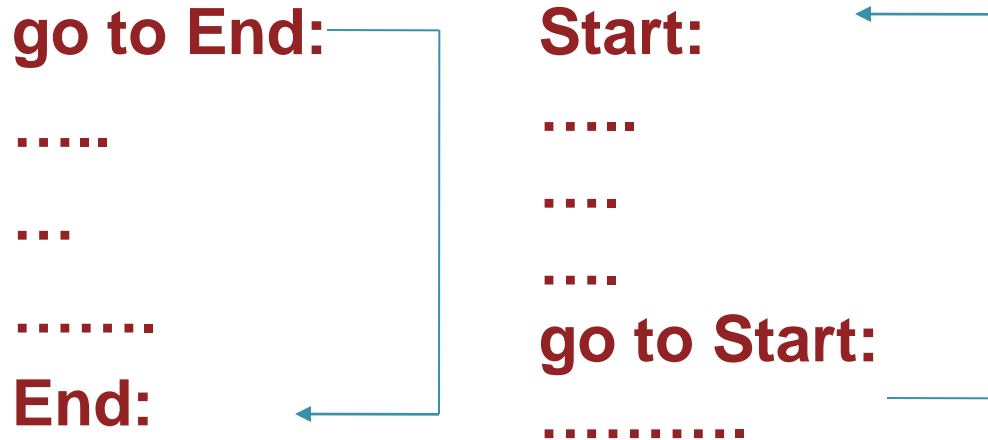
This control structure provides the unconditional control statement means that using this control can move from any where in the program with any condition checking

The syntax of go to statement is follows:

go to label ;

- Where go to is a keyword**
- Label is a symbolic constant written either in a upper case or lower case**

Sequence Control b/w Statements:



Break Statement:

It is used to terminate the loop and exit from a particular switch case label. by using this control structure suddenly comes out from the currently accessing loop and continues with the next statement .

The syntax of break is as follows:

```
..
  break;
...

```

Sequence Control b/w Statements:

Continue Statement:

This is used to skipping a part of loop. The control does not come out of the loop instead it skips the remaining statements within the body of that loop and transfer to the beginning of the loop. Thus a **break** statement will exit a loop will **continue** will repeat the loop again The syntax is as follows:

```
...  
continue;  
...
```

Structured Sequence Control:

Most of language provides a set of control statements for iteration, decision control structure known as structured sequence control. The following the some points that are related with structured sequence control:

❑ **Avoid the unstructured control structure like go to**

Sequence Control b/w Statements:

Will make the programming style as unstructured. The unstructured programming style have a lot of drawbacks as:

1. **More chance of programming errors**
2. **It's quite hard to fix bugs**

That's why unstructured control(i.e. go to) should be avoided)

- ❑ **Single entry and single exit must be implement**
- ❑ **Multiple path must be avoided**

2.Compound Statement:

A compound statement is a sequence of statements that may be treated as single statement in the construction of larger statements. The syntax of compound are:

Begin

.... **--Sequence of statement(one or more)**

End

The order in which they appear in memory determines the order in which they will be executed.

3. Control Statement:

The control statements used to control the execution of the instruction in a different manner instead of the sequential way also may perform special tasks like:

- ❑ **Select a set of statements from several alternatives (if else, if else then, Nested if, Switch Case etc.)**
- ❑ **Skip the certain statement or in some condition continue the statement (break/continue)**
- ❑ **Repeat a set of instructions after or before checking a condition (for loop, do while, while do, repeat until)**
- ❑ **Unconditional jump that is totally defined by the programmer (goto statement)**

The example is given by:

```
If (a>5)
{
...
...
}
Else
{
...
}
```

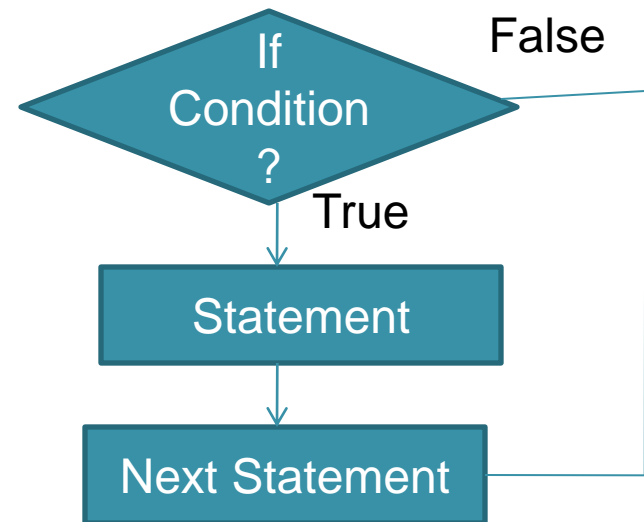
4. Conditional Statement:

Some conditions it is required to check the conditions to make a decision. This involves performing a logical test. This results in either true or false. The conditional execution involves both decision making and branching. Conditional statements involves the following:

- ❖ **If -statement**
- ❖ **If-else statement**
- ❖ **Nested-if-statement**
- ❖ **Switch statement**

a)If-statement

```
if (condition)
{
    statement;
}
```



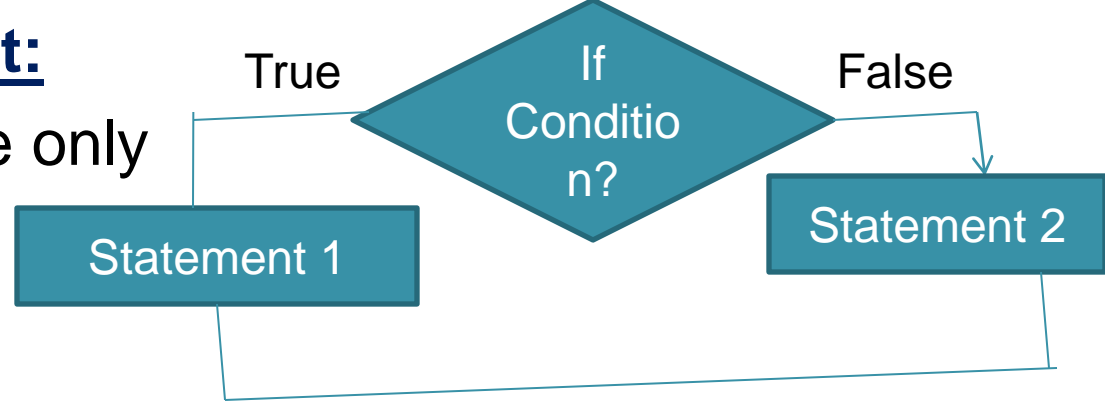
b)If-else-Statement:

It is used to execute only one action.

The example :

If (a>5) condition

```
{  
Statement 1;  
}  
Else  
{  
Statement 2;  
}
```

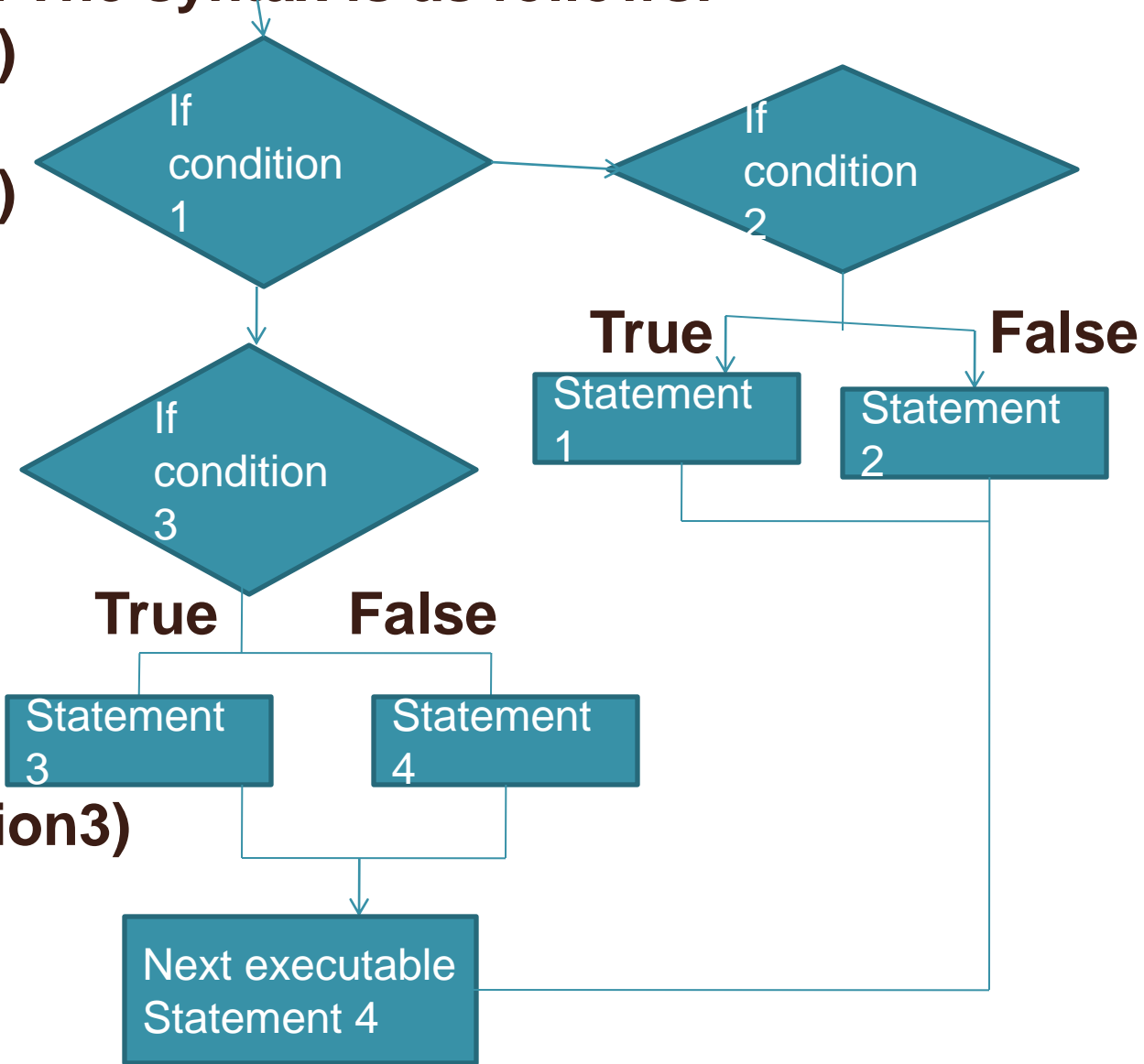


c)nested-if-Statement:

If there are more than two alternatives then nested-if-statement are

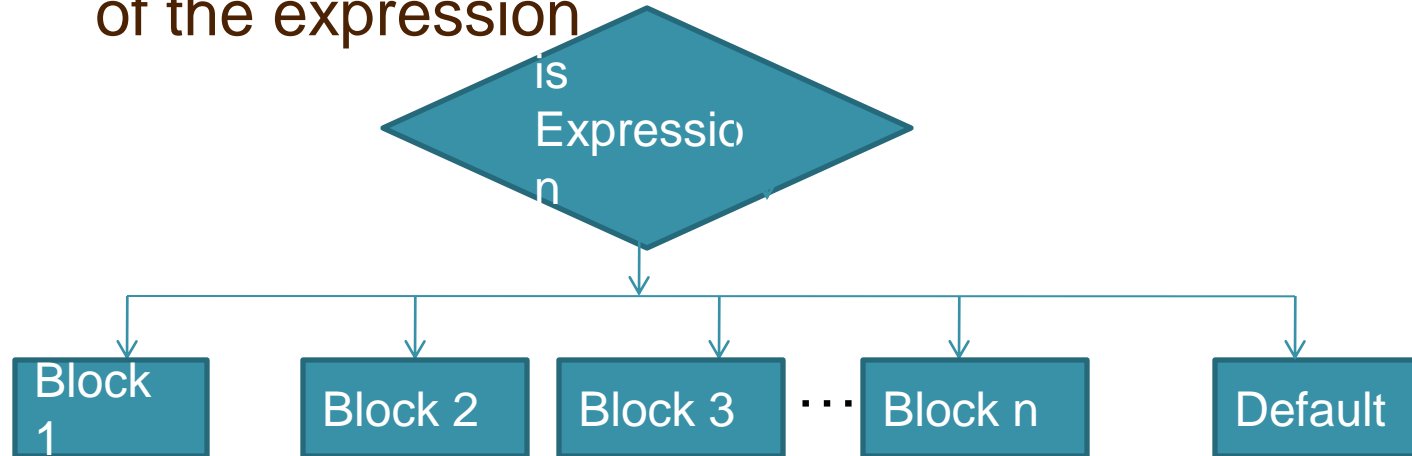
Implemented. The syntax is as follows:

```
if (condition1)
{
if (condition2)
{
statement1;
}
Else
{
Statement2;
}
}
else if(condition3)
{
statement3;
}
else
{ statement}
```



d) Switch Statement:

The switch statement is used to use multi-way branching. It means it provides a way for a programmer to select any one of the several alternatives depending on the value of the expression.



Sequence Control b/w Statements:

2(a) For Statement:

This is the iterative statement. The body of the loop will execute till condition is true. The block of for loop is given below:

```
For (a=1;a<=10; a++)
```

```
{
```

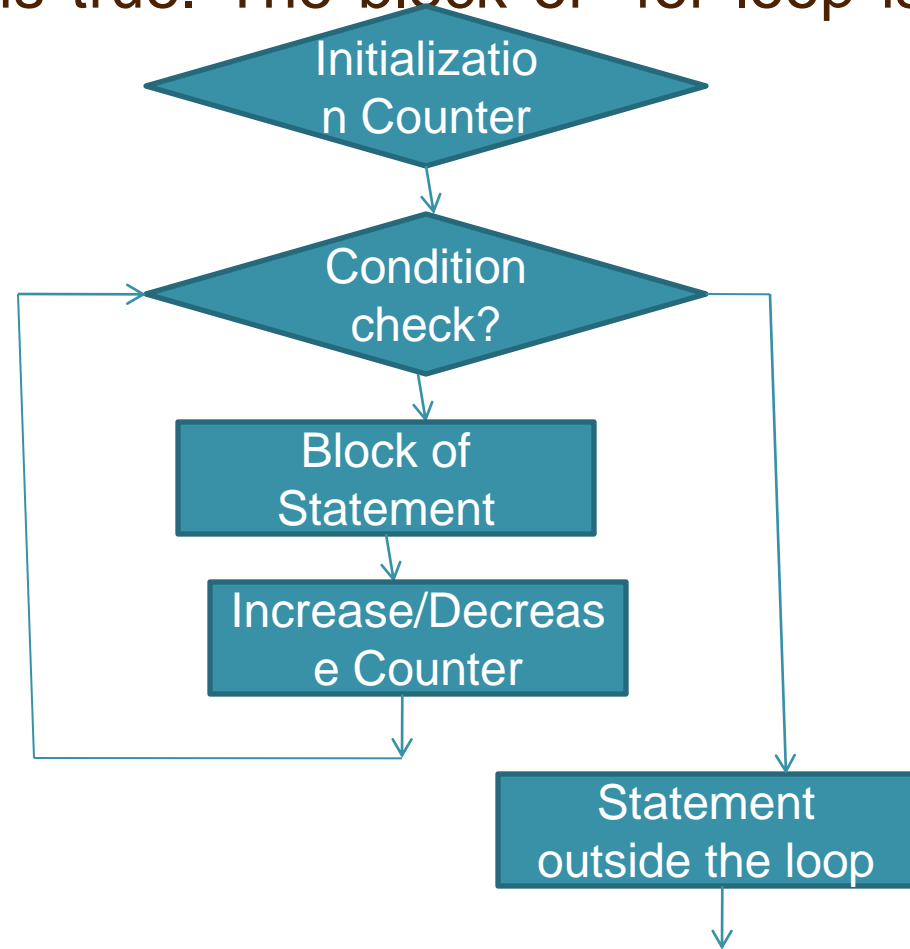
```
Statement;
```

```
....
```

```
...
```

```
....
```

```
}
```



2(b) While Statement:

This is also a interactive statement The body of the loop will execute till condition is true. The block is given below:

```
A=1;
```

```
While(a<=10)logical expression
```

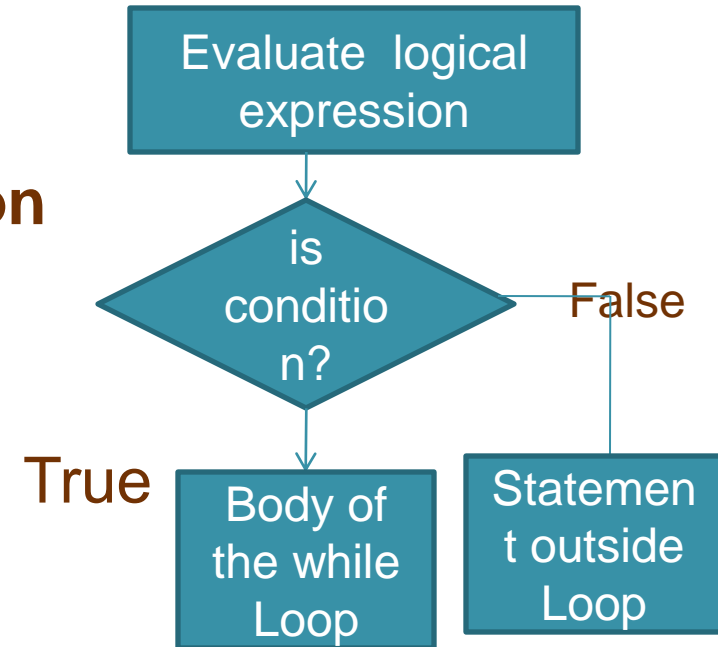
```
{
```

```
..
```

```
...
```

```
a++;
```

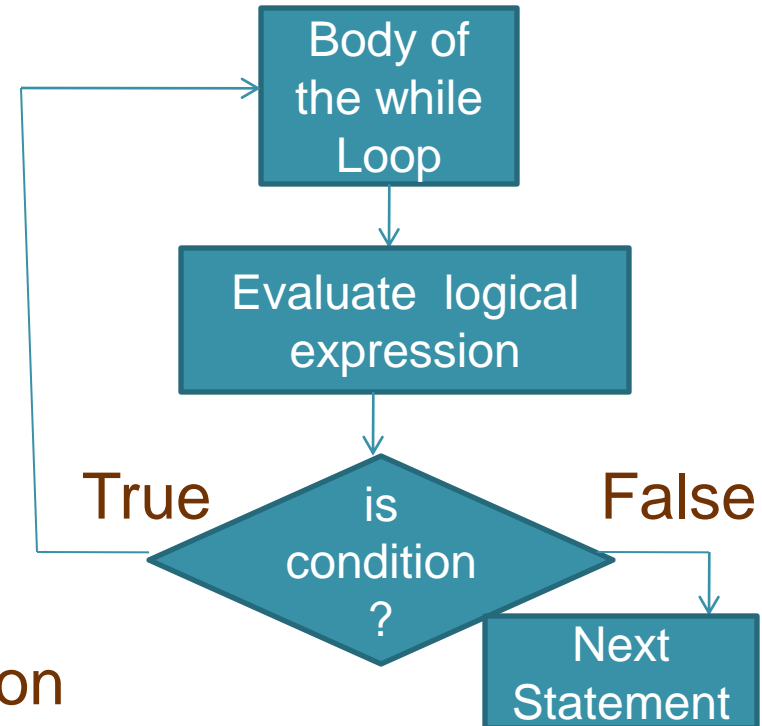
```
}
```



2(c) Do while Statement:

This is also a interactive statement The body of the loop will execute till condition is true. The block is given below:

```
A=1;  
Do  
{  
..  
..  
a++;  
}while (a<=10);logical expression
```



Assignment

- Explain sequence control in statement