L-5 FLOW OF CONTROL

Expt. No..... Page No.....

Statements :- Instructions given to comp. to perform any kind of action. - data movement,
- making decision
- repeating action

Types of Statement : → Empty → Simple → Compound

EMPTY: statement which does nothing

>>> pass

- null op" statement
- It does nothing & moves to next statement
- Use of Pass : where syntax of language requires presence of statement but where logic of prog. does not. Ex: loops.

SIMPLE: Executable statement is simple statement

```
>>> name = input("Enter ")
>>> print(name)
```

COMPOUND: grp of statement executed as unit

Syn :- <compound statement header>:
<body>

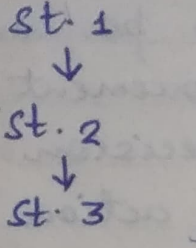
header : begins with keyword & ends with a colon.

body :- 1 or more statement
- all statement are having same indentation.

Statement Flow Control : → sequence → selection → iteration

Date ..... Teacher's Signature : .....

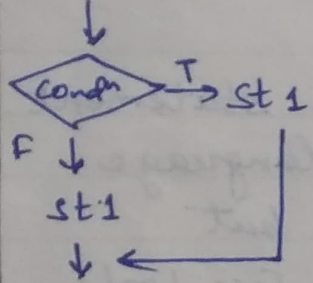
SEQUENCE :- means statements executed sequentially
- normal flow of control in prog. & is simplest one



WORKING:
Python prog. begin with first st. → Each st is executed → When final statement of prog. is executed prog. is done.

SELECTION :- means execution of st. depending upon cond^n test

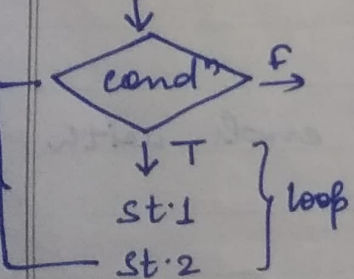
[IF St.]
- Also called decision construct ∵ it helps in making decision about which set of instruction is to be executed.



WORKING:
If Py. prog. cond^n evaluates to true a set of statement is followed, if false another set of st. is followed.

REPETITION / ITERATION / LOOPING:

[FOR/WHILE] St.
- means repeatition of set of st. depending upon cond^n test
- set of st. that are repeated again and again is called body of loop.
- cond^n on which execution / exit of loop depends is called test-cond^n / exit cond^n



WORKING:
In Py. prog. till the time cond^n is True, a set of st. are repeated
As soon as cond^n becomes false, repeatition stops

Two Types: Cond^{n}al Loop : while (cond^n based loop)
Counting loop : for (loop for given no. of times)

Expt. Page No.

IF STATEMENT :- It test a cond^n & if cond^n evaluates to True, it carries out some instruction & does nothing in case cond^n evaluate to false.

(one cond^n)

Syn:-

```
if <cond^n exp>:
    st.
    st.
```

} If its True, st. of body is executed otherwise ignored.

Ex:
```
ch = input ("Enter a character")
if ch == ' ':
    print ("You entered a space")
```

IF - ELSE STATEMENT :- If evaluates True, block below if gets executed.
If evaluates false, block below else gets executed.

(2 cond^n) can be check.

Syn :-
```
if <cond^n exp.>:
    st 1
else:
    st 1.
```

Ex:
```
if a>=0 :
    print (a)
else:
    print (a, " is a negative number")
```

IF - ELIF - STATEMENT :- If inside another if-else.
- as many times possible u can add elif

Syn :-
```
if <cond^n exp>:
    st.
elif <cond^n exp>:
    st.
```

& if <cond^n exp>:
```
    st.
elif <cond^n exp>:
    st.
else:
    st.
```

Date _____ Teacher's Signature : _____

Ex:
```
if runs >= 100:
    print("century")
elif (runs>=50):
    print ("fifty")
else:
    print (" Score less")
```

NESTED IF STATEMENT :- if inside another if
(Multiple if cond") if inside another elif
if inside another else body.

Syn:-

1)
```
if <cond">:
    if <cond">:
        st.
    else:
        st.
else:
    st.
```

2)
```
if <cond">:
    st.
elif <cond">
    if <cond">
        st.
    else:
        st.
else:
    st.
```

3)
```
if <cond">
    st.
elif <cond">
    st.
else:
    if <cond">
        st.
    else:
        st.
```

4)
```
if <cond">:
    if <cond">:
        st.
    else:
        st.
elif <cond">:
    if <cond">:
        st.
    else:
        st.
else:
    if <cond">:
        st.
    else:
        st.
```

STORING_COND" :

We can name the cond" as sometimes cond" used in code are complex and repetitive.

Ex: if deposit<2000 & time>=2:
        st:

Named cond" :- a=deposit<2000 & time >=2

Ex :-
```
if a:
    st.
```

Range fn : - used with for loop.
Expt. No_____ - generates list which is special sequence type. Page No_____

Sequence :- succession of values bound together by a single name. Ex: String, list, tuple

Syn:- range (<lowerlimit>, <upper limit>, <step value>)
range (l, u, s)

Ex:- range (10) : (0, 1, 2, 3, 4, 5, 6, 7, 8, 9)
range (1, 5) : (1, 2, 3, 4,)
range (1, 5, 2) : (1, 3)

FOR LOOP :- process the item of any sequence like list or string, one by one.

Syn:- for <variable> in <seq> :
st.

Ex:- for a in [1, 2, 3]:
print (a)

O/P - 1
2
3

WORKING:
→ Loop variable is assigned 1st value in seq.
→ All st. in body of for loop executed with assigned value of loop.
→ Then next value is assigned
→ Continues till all values in seq. are processed.

Ex:- for a in range (1, 10, 2):
print (a)

O/P 1
3
5
7
9

Range fn used for larger list

NESTED LOOP :- LOOP Inside another loop.

Syn:- for i in range (1, 10, 2):
for j in range (1, i):
print ("*")
print ()

Working:-
→ Inner loop is executed
→ i value is assigned (1, 1) (1, 3)
→ value of outer loop (1, 5) will change only after inner loop is completely finished

Date _____ Teacher's Signature : _____

Range fn : - used with for loop.
Expt. No _____ - generates list which is special sequence type.

Page No _____

Sequence :- succession of values bound together by a single name. Ex: String, list, tuple

Syn:- range (<lower limit>, <upper limit>, <step value>)
range (l, u, s)

Ex:- range (10) : (0, 1, 2, 3, 4, 5, 6, 7, 8, 9)
range (1, 5) : (1, 2, 3, 4,)
range (1, 5, 2) : (1, 3)

FOR LOOP :- process the item of any sequence like list or string, one by one.

Syn:- for <variable> in <seq> :
st.

Ex:- for a in [1, 2, 3]:
print (a)

O/P - 1
2
3

WORKING:
→ Loop variable is assigned 1st value in seq.
→ All st. in body of for loop executed with assigned value of loop.
→ Then next value is assigned
→ Continues till all values in seq. are processed.

Ex:- for a in range (1, 10, 2):
print (a)

O/P 1
3
5
7
9

Range fn used for larger list

NESTED LOOP :- LOOP Inside another loop.

Syn:- for i in range (1, 10, 2):
for j in range (1, i):
print ("*")
print ()

Working:-
→ Inner loop is executed
→ i value is assigned (1, 1) (1, 3)
→ value of outer loop will change only after inner loop is completely finished (1, 5)

Date _____ Teacher's Signature : _____