

2. | 30/7/25 | Implement conditional, control  
| and looping statements | 15 |  
| | 13/14/25 |

# Design algorithm easiest condition to design

integers Design easy to find fully result of all numbers  
 first search across row then column of first row  
 then move to next row and so on

## Output:

Enter the room temperature : 20

comfortable room with moist, avoid fast sweat top A

(heat) effect the sweating system  
 (heat) effect the sweating system  
 (heat) effect the sweating system  
 (heat) effect the sweating system

affection with subject  
 affected with subject  
 affected with subject

((if fast not wet then (heat) sweating tri = 1 else  
 ((if fast & not sweating then (heat) sweating tri = 2 else  
 ((if fast & sweating then (heat) sweating tri = 3 else  
 heat effect 3 < heat effect 2 < heat effect 1  
 or heat effect 2 < heat effect 1  
 or heat effect 3 < heat effect 1  
 if heat effect 3 < 0))

55/5

AER TECH - CSE	
Ex No.	
12	PERFORMANCE (2)
12	RESULT AND ANALYSIS (2)
12	CODE (2)
12	OB (2)
12	TEST (2)

8/12

base on maximum design conditions  
 first no load condition first (100% of maximum power)

Mar 38

3/7/25 Task -2: Implement conditional, control and looping statements

(a) Temperature Alert System (if - elif):

Aim:-

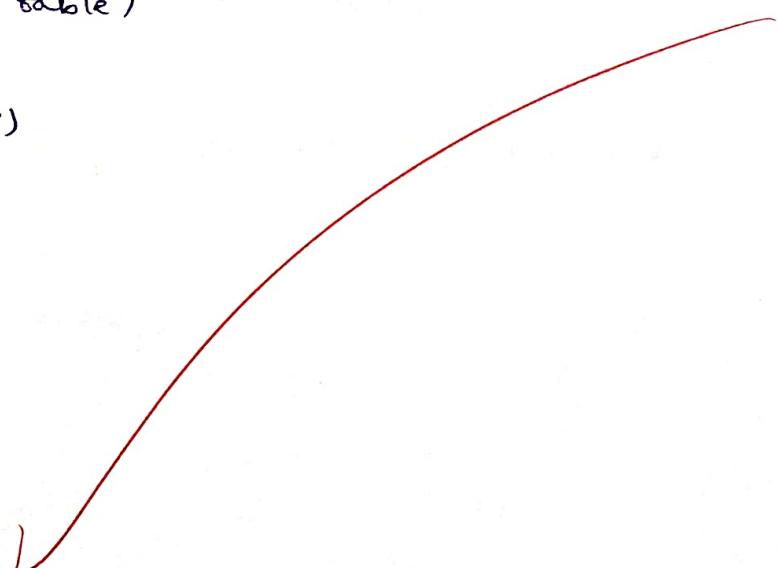
To write a python program that classifies temperature as "Too cold", "Comfortable", or "Too Hot" using conditional (if - elif - else) statements.

Algorithm :-

1. Start
2. Accept temperature input from the user.
3. If temperature < 18, print "Too cold".
4. Else if temperature is between 18 and 25 (inclusive) print "Comfortable".
5. Else, print "Too Hot".
6. End.

Program :-

```
# Temperature Alert System
temperature = float(input("Enter the room temperature:"))
if temperature < 18:
    print("Too cold")
elif 18 <= temperature <= 25:
    print("Comfortable")
else:
    print("Too Hot")
```



Result:-

The program correctly identifies the room temperature range and prints an appropriate alert.

✓ 09

Program to check if the password is correct or not

if the password is correct then print "Access Granted"

(else) print "Wrong password"

### Output:-

Enter password: testmoring soft 19. as first 10

Incorrect password. Try Again. else can be checked

Enter password: admin

Incorrect password. Try Again.

Enter password: admin123

Access Granted.

## 07/25 b. Password Retry System (While Loop)

### Aim:-

To implement a password retry system using a while loop that allows a maximum of 3 attempts.

### Algorithm:-

1. Start
2. Set correct password as "admin123".
3. Set attempt counter to 0.
4. While attempts < 3:
  - Ask the user for password input
  - If correct, print "Access Granted" and exit loop.
  - Else, increment attempts and print "Try Again".
5. If attempts == 3, print "Access Denied".
6. End.

### Program:-

```
#password Retry system
correct_password = "admin123"
attempts = 0

while attempts < 3:
    entered_password = input("Enter password:")
    if entered_password == correct_password:
        print("Access Granted")
        break
    else:
        print("Incorrect password. Try Again")
        attempts += 1
if attempts == 3:
    print("Access Denied")
```

### Result:-

The program allows upto 3 attempts and grants or denies access successfully.

(good start) with 1/10 marks

To solve a given problem first breakdown in framework  
as per the 3 steps mentioned in slide

### Output:-

Enter a number: 5

Factorial of 5 is "120" having 3 digits

so it needs 3 digits  
so we can take 3 digits  
right boundary is 324  
left boundary is 100  
and thus "5! is 120"  
length of 3 digits fits requirement

"length 3 digits" since  $2^3 = 8$  digits

not for perfect boundary

"2 is sufficient" resulting from

$2^3 = 8$  digits

so it is written

("initialising" perfect right boundary - boundary)

initialising - max is 324 - boundary - largest x

(second largest 120) thing

the end

("empty" left boundary - second left) thing

not important

so it is ignored

("initial 324") thing

so it is written

### C. Factorial Finder (for loop)

Aim:-

To write a python program that calculates the factorial of a number using a for loop.

Algorithm :-

1. Start
2. Input a number from the user
3. Initialize result as 1.
4. Loop from 1 to number (inclusive), multiply result by each value.
5. Print the result.
6. End

Program :-

```
# Factorial Finder
Number = int (input ("Enter a number :"))
factorial = 1
for i in range (1, number + 1):
    factorial *= i
print ("Factorial of ", number, "is", factorial)
```

VEL TECH - CSE	
EX NO.	2
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
AVA VOCE (5)	5
GRD (5)	15
	61

83/6

Result:-

The program correctly calculates the factorial of the entered number using a for loop.