

11/08

Task: Implement conditional, control and looping statements.

Aim: To implement conditional, control and looping statements using Python.

Q.1 You are developing a simple grade management system for a school. The system needs to determine the grade of a student based on their score in a test. The grading system follows these rules:

If the score is 90 or above, the grade is "A".

If the score is bw 80 and 89, the grade is "B".

If the score is bw 70 and 79, the grade is "C".

If the score is bw 60 and 69, the grade is "D".

If the score is below 60, the grade is "F".

Algorithm:

1. start

2. Get the input mark from the user.

3. with the use of an If-elif-else statement do.

→ If the marks  $\geq 90$ , Print grade "A".

→ If the mark is bw 80 and 89, Print grade "B".

→ If the mark is bw 70 and 79, Print grade "C".

→ If the mark is bw 60 and 69, Print grade "D".

→ If the mark is below 60, Print grade "F".

4. stop.

Program:

```
score = int (input ("Enter the score:"))
```

```
if score >= 90:
```

```
    print ("The Grade is A")
```

```
elif (score <= 89 and score >= 80):
```

```
    print ("The Grade is B")
```

```
elif (score <= 79 and score >= 70):
```

```
    print ("The Grade is C")
```

```
elif (score <= 69 and score >= 60):
```

```
    print ("The Grade is D")
```

```
else:
```

```
    print ("The Grade is F")
```

Output: for zero soft stations of choices D or E  
~~= = = = = RESTART; C: \user~~

Enter the score: 60

The Grade is D.

~~zero soft limit of memory reading is given or not  
 so which went to the external soft meter if present or to  
 alarm of memory limit giving reading~~

GRADE	TEST
1	(0)
2	(2)(2) EOKANOL
3	(2) E12YUANAQAL
4	EUNAQUAM EUNAV
5	2YICE (2)(2) OJ
6	(0)
7	1. EKAMATOC

8/21/03

two buttons of which ever was memory soft limit though  
 two buttons were both buttons were zero soft

- Q.2: The electronics maintenance team at a data center needs a tool to assess the health status of UPS backup batteries based on their current charge percentage. You are asked to develop a Python program that accepts the battery charge percentage as input and categorizes the battery health using the following conditions:
- If the percentage is greater than, or equal to 90, display: "Excellent Battery Health".
  - If the percentage is between 70 and 89, display: "Good Battery Health".
  - If the Percentage is between 40 and 69, display: "Average Battery Health".
  - If the percentage is between 40, display: "Poor Battery Health".

Task:  
write a python program that: uses ladderized if-elif-else statements.

Algorithm:

1. Accept battery Percentage from the user.
2. use ladderized if-elif-else to determine the health category:
  - If Percentage  $\geq 90 \rightarrow$  "Excellent Battery Health".
  - If  $70 \leq \text{Percentage} < 90 \rightarrow$  "Good Battery Health".
  - If  $40 \leq \text{Percentage} < 70 \rightarrow$  "Average Battery Health".
  - If  $\text{Percentage} < 40 \rightarrow$  "Poor Battery Health".

frigool bno lantos, tonofibros fasmelgmi  
Hematite

## Python Program:

```
# Battery Health checker.  
Percentage = int(input("Enter battery Percentage :"))  
if Percentage >= 90:  
    print("Excellent Battery Health")  
elif Percentage >= 70:  
    print("Good Battery Health")  
elif Percentage >= 40:  
    print("Average Battery Health")  
else:  
    print("Poor Battery Health")
```

Input: Battery charge P% Percentage (integer)  
Output: Enter battery Percentage : 85  
Good Battery Health.

("85 is not integer") file : 85  
: op = < 85  
("A is short str") file  
:(68 - < 85) bne P8 -> 85 file  
("8 is short str") file  
: (op = < 85) bne P7 -> 85 file  
("3 is short str") file  
:(68 - < 85) bne P6 -> 85 file  
("0 is short str") file  
("7 is short str") file

Sample Input:

Enter height of visitor 1 in cm : 130 last n aban  
enter height of visitor 2 in cm : 110 no belief in culture  
enter height of visitor 3 in cm : 150 foolish of history  
Enter height of visitor 4 in cm : 90 spreads jotted  
Enter height of visitor 5 in cm : 125. jotted off

Sample Output:

"At least 100cm tall to enter" →  
"At least 120cm tall to enter" →  
"At least 150cm tall to enter" →  
"At least 90cm tall to enter" →  
"At least 110cm tall to enter" →

Allowed  
Not allowed  
Allowed  
Not allowed  
Allowed.

2.3 You're coding a system at an amusement park that checks the height of each visitor.

- If the height is 120 cm or more, print "Allowed".  
→ otherwise, print "Not allowed".  
Repeat this for 5 visitors.

## Algorithm:

1. start the program
  2. set the total number of visitors to 5.
  3. Loop from visitor 1 to visitor 5.
    - Accept the height of the visitor as input (integer).
    - If height is greater than or equal to 120,
    - Print "Allowed".
    - Else, print "Not allowed".
  4. End the loop, after 5 visitors have been checked.
  5. stop the program.

Program:

```

for i in range (1,6):
    height = int (input ("Enter height of visitor (in cm):"))
    if height >= 120:
        print ("Allowed to ride")
    else:
        print ("Not allowed to ride")

```

else:

print ("Not allowed to sit")

VET TECH - ESE	
EX NO.:	✓
PER NO.:	✓
PERFORMANCE (3)	✓
RESULT AND ANALYSIS (3)	✓
VIVA VOCE (3)	✓
RECORD (3)	✓
FATL (20)	✓
EW WITH DATE	✓

Result: Thus, the Python program was successfully implemented using conditional statements (if-else), control flow and looping statements.

right  
rint  
for 5 visitors  
is 120  
"Not allowed".  
total number of visitors to 5.  
in visitors 1 to visitor 5:  
at the height of the visitors as input (initial)  
height is greater than or equal to 120,  
print "Allowed".  
else, print "Not allowed".  
loop, after 5 visitors have been checked.  
the program ends.

in range (1, 6):  
= int(input("Enter height of visitor {i} in cm:"))  
height >= 120:  
print("Allowed to ride.")  
else:  
print("Not allowed to ride.")

X NO.	PERFORMANCE (%)	RESULT AND ANALYSIS (%)	VIVA VOCE (%)	RECORD (%)	TOTAL (%)	WITH DATE
1	2	5	5	5	7	2023-01-25/08
2	5	5	5	5	7	2023-01-25/08
3	5	5	5	5	7	2023-01-25/08
4	5	5	5	5	7	2023-01-25/08
5	5	5	5	5	7	2023-01-25/08
6	5	5	5	5	7	2023-01-25/08

~~Result: Thus, the python program was successfully implemented using conditional statements (if-else), control flow and looping statements.~~