Date:

Task:2 Implement Conditional, Control and looping Statement

2.1

Aim: To implement conditional, control and looping statements using python 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 between 80 and 89, the grade is "B".

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

If the score is between 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 >= 90 Print grade "A".
   * If the mark is between 80 and 89 print grade "B".
   * If the mark is between 70 and 79 print grade "C".
   * If the mark is between 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:

Enter the score : 60

The Grade is D

Result: To implement Conditional, control and looping statement for simple grade Management system using Python code successfully executed.

Date: 13/8/25 Task: 2.2

Aim: Implement Conditional, Control and looping Statements

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 programming 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 below 40, display: > "Poor Battery Health"

Algorithm:

1. Accept battery percentage from the user.
2. Use ladderized if-elif-else to determine the health category:

* If percentage ≥ 90 → "Excellent Battery Health"
* If 70 ≤ percentage < 90 → "Good Battery Health"
* If 40 ≤ percentage < 70 → "Average Battery Health"
* If percentage < 40 → "Poor Battery Health".

Python Program:

```
# Battery Health Checker
Percentage = int(input("Enter battery percentage:"))
if percentage >= 90:
    print("Excellent Battery Health")
```

Input:

Battery charge percentage (integer)

Sample output:

Enter battery percentage : 85
Good Battery health.

```
elif Percentage >= 70:
    Print (" Good Battery Health")
elif  Percentage >= 40:
    Print(" Average Battery Health")
else :
    Print(" Poor Battery Health")
```

Result: To implement conditional, control and looping statement using python for battery charge health is successfully executed.

Date:

Task-2.3

Aim: Implement Conditional, control and looping statements

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

* If the height is 120cm 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 (in cm).
   * 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(f"Enter height of visitor {i} in cm:"))
    if height >= 120:
        Print("Allowed to ride.")
    else:
        Print("Not allowed to ride.").
```

Sample Input:

Enter height of visitor 1 in cm: 130
Enter height of visitor 2 in cm: 110
Enter height of visitor 3 in cm: 150
Enter height of visitor 4 in cm: 90
Enter height of visitor 5 in cm: 125

Output:

Allowed
Not allowed
Allowed
Not allowed
Allowed.

VEL TECH - CCE	
EX NO.	2
PERFORMANCE (5)	5
RESULT AND VIVA (5)	5
VIVA VOCE (5)	5
RECORD (5)	
TOTAL (15)	15
SIGN WITH DATE	

Result: The python program was successfully implemented using conditional statements, control flow and looping statements.