

Task 7:- Utilizing 'Functions' Concepts in Python Programming.

Aim: To write the Python Program using Functions' Concepts in Python Programming.

Problem 7.1:-

You are developing a small Python script to analyze and manipulate a list of student grades for a class project. Write a Python program that satisfies the above requirements using the built-in functions `print()`, `len()`, `type()`, `min()`, `max()`, `sorted()`, `reversed()`, and `range()`.

Algorithm:

1. Start the Program.
2. Print a welcome message & outputs a simple grading.
3. Determine & print the number of students. Uses `len()` to find the no. of elements in the Student names list.
4. Print the types of lists - uses `type()` to show the type of student names.
5. Find & print highest and lowest grades: uses `max()` and `min()` to determine the highest & lowest values in student grades.
6. Printed sorted list & grades: uses `sorted()` to sort the grades.
7. Print reversed list of grades: uses `reversed()` to reverse sorted list & converts it to a list.
8. Generates and print a range of grade indices:

uses range() to create a list of indices from 1 to the no. of Students.

9. Stop.

Program:-

```
def analyze_student_grades(j):
    # Sample data
    Student_names = ['Alice', 'Bob', 'Charlie', 'Diana']
    Student_grades = [85, 92, 78, 90]
```

#1. Print a welcome message

```
Point("Welcome to Student Grades Analyze! \n")
```

#2. Determine & print the number of Students

```
num_students = len(Student_names)
```

```
Point("Number of Students: ", num_students)
```

#3. Print the type of Student names list & grades list

```
Point("In Type of Student_names list: ", type(Student_names))
```

```
Point("Type of Student_grades list: ", type(Student_grades))
```

#4. Find & print the highest & lowest grades

~~highest_grade = max(Student_grades)~~~~lowest_grade = min(Student_grades)~~~~Point("In Highest grade: ", highest_grade)~~~~Point("Lowest grade: ", lowest_grade)~~

#5. Print the list of grades sorted in ascending order

```
Sorted_grades = sorted(Student_grades)
```

```
Point("In Sorted grades: ", Sorted_grades)
```

Output :-

Welcome to the Student Grades Analyzer!

Number of Students : 4

Type of Student names list : < class list >

Type of Student grades list : < class list >

[OP 185, SP, DA] = 29h0B - habop

Highest grade : 92

lowest grade : 78

Sorted grades : [78, 85, 90, 92]

Reversed grades : [92, 90, 85, 78]

Grade indices from 1 to number of students : [1, 2, 3]

29h0B 3 tei 29mon tribut2 70 98t 9it. tribut2 8t

tribut2 9it, " tei 29mon tribut2 70 98t 9it" tribut2 8t

(29b - tribut2) 9it, " tei 29mon tribut2 70 98t 9it" tribut2 8t

(29b - tribut2) 9it, " tei 29mon tribut2 70 98t 9it" tribut2 8t

29h0B tribut2 3 tribut2 9it tribut2 8t

(29h0B - tribut2) 9it = 29h0B - tribut2 8t

(29h0B - tribut2) 9it = 29h0B - tribut2 8t

(29h0B - tribut2) 9it = 29h0B - tribut2 8t

(29h0B - tribut2) 9it = 29h0B - tribut2 8t

29h0B tribut2 3 tribut2 9it tribut2 8t

(29h0B - tribut2) 9it = 29h0B - tribut2 8t

(29h0B - tribut2) 9it = 29h0B - tribut2 8t

#6. Print the list of grades in reverse order.
reversed_grades = list(reversed(sorted_grades))
print("Reversed grades:", reversed_grades)

#7. Generate & print a range of grade indices from 1 to the number of students

grade_indices = list(range(1, num_students + 1))
print("In Grade indices from 1 to no. of students:", grade_indices)

Run the analysis
analyze_student_grades[].

Problem 7.2 :-

You are tasked with creating a small calculator application to help users perform basic arithmetic operations and greet them with a personalized message. Your application should perform the following tasks: addition, subtraction, multiplication, division.

Algorithm :-

1. Start the program.
2. User input for numbers: The program prompts the user to enter two numbers.
3. User input for operations: The program prompts the user to choose an arithmetic operation.
4. Perform operation: Based on the user's choice, the program performs the chosen arithmetic operation using the defined functions.
5. Display result: The program displays the result of the operation.
6. STOP.

7.2 Program :-

```
def add [a,b]:  
    """Return the sum of two numbers.  
    Return a+b  
def subtract [a,b]:  
    """Return the difference between two numbers.  
    Return a-b  
def multiply [a,b]:  
    """Return the product of two numbers.  
    Return a*b  
def divide [a,b]:  
    """Return the quotient of two numbers. Handles  
    division by zero.  
    if b!=0:  
        return a/b  
    else:  
        return "Error: Division by zero"  
def greet (name):  
    """Return a greeting message for the user.  
    return f"Hello, {name}! Welcome to Program."  
def main []:  
    # Demonstrating the use of user-defined functions  
    # Arithmetic operations  
    num1=10  
    num2=5  
    print ("Arithmetic operations:")  
    print (f"Sum of {num1} & {num2}:", add(num1,num2))  
    print (f"Product of {num1}, {num2}:", multiply(num1,num2))  
    # Greeting the user
```

Output :-

Arithmetic operations :

Sum of 10 and 5 : 15

Difference between 10 and 5 : 5

Product of 10 and 5 : 50

Quotient of 10 and 5 : 2.0

Greeting :

Hello, Alice ! welcome to the program.

```
User_name = "Alice"  
print ("In Greeting:")  
print (greet(user_name))
```

```
# Run the main function  
if __name__ == "__main__":  
    main()
```

Qb

VEL TECH	
L.K NO.	
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	6
VIVA VOCE (5)	4
RECORD (5)	4
TOTAL (20)	18
SIGN / IT DATE	08/20

Result Thus, the Python program using Functions Concept was successfully executed and the output was verified.

