

Task No: 6 Utilizing 'functions' Concepts in python programming

Date: 03/9/25

- Ques: 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 requirement using the build-in function printf(), len(), type(), max(), min(), sorted(), reversed() and range().

Algorithm:-

1. Start the program.
2. print a welcome program: outputs a simple greeting.
3. Determine and print the number of students: uses len() to find the number of elements in the student_names list.
4. Print the type of lists: uses type() to show the type of the Student_names and Student_grades lists.
5. Find and print highest and lowest uses max() and min() to determine the highest and lowest values in Student_grades.
6. Print sorted list of grades: uses sorted() to sort the grades.
7. Print reversed list of grades : user reversed() to reverse the sorted list and converts it to a list.
8. Generate and print a range of grades indices: use range() to Create a list of indices from 1 to the number of students.
9. Stop.

Program:-

```
def analyze_student_grades():  
    # Sample data  
    Student_names = ["Alice", "Bob", "Charlie", "Diana"]  
    Student_grades = [85, 92, 78, 90]  
    # Print a welcome message  
    print("Welcome to the student Grades analyze!\n")  
    # 2. Determine and print the number of students  
    num_students = len(Student_names)  
    print("Number of Students:", num_students)  
    # 3. Print the type of the student names list and the grade  
    list  
    print("In Type of Student-names list:", type(Student_names))
```

```
print("Type of student-grades list", type(student-grades)),
```

4 Find and print the highest and lowest grades

```
highest-grade = max(student-grades)
```

```
lowest-grade = min(student-grades)
```

```
print("In highest grade:", highest-grade)
```

```
print("lowest grade:", lowest-grade)
```

5. print the list of grades Sorted in ascending order

```
Sorted-grades = sorted(student-grades)
```

```
print("In sorted grades:", reversed-grades)
```

6. print the list of grades in reverse order

```
reversed-grades = list(reversed(sorted-grades))
```

```
print("Reversed grades", reversed-grades)
```

7. Generate and print range of grade indices from 1 to the number of students

```
grade-indices = list(range(1, num-student + 1))
```

```
print("In Grade indices from 1 to number of students:", grade-indices)
```

Run the analysis

```
analyze-student-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' >

highest grade : 92

lowest grade : 78

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

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

grades indices from 1 to number of students : [1, 2, 3, 4]

If you are asked 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 task: addition, subtraction, multiplication and division.

Algorithm:-

1. Start the program
2. User input for Numbers: The program prompts the user to enter two numbers.
3. User input for operation: The program prompts the user to choose an arithmetic operation (addition, subtraction, multiplication, division).
4. Perform operation: Based on the user 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.

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 the program!"
```

output :-

arithmetic operations.

Sum of 10 & 5 : 15

Difference b/w 10 & 5 = 5

Product of 10 and 5 = 50

Quotient of 10 and 5 : 20

Greeting :

Hello. Alice: welcome to the program

```

def main():
    # Demonstrating the use of user-defined functions
    # Arithmetic operations
    num1 = 10
    num2 = 5
    print("Arithmetic operations:")
    print(f"Sum of {num1} and {num2}:", add(num1, num2))
    print(f"Difference between {num1} and {num2}:", subtract(num1, num2))
    print(f"Product of {num1} and {num2}:", multiply(num1, num2))
    print(f"Quotient of {num1} and {num2}:", divide(num1, num2))

    # Greeting the user
    user_name = "Alice"
    print("In Greeting:")
    print(greet(user_name))

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

```

VEL TECH	
EX NO.	6
PERFORMANCE (5)	S
RESULT AND ANALYSIS (5)	S
VIVA VOCE (5)	S
RECORD (5)	S
TOTAL (20)	15
DATE / WITH DATE	

Result:- Thus, the python program using 'functions' concepts was successfully executed and the output was verified.