Task No: 6 utilizing 'Function' concepts in python programming

11/9/25

Aim: To write python program using 'Function' concepts in python programming

Algorithm:

1. Start the program
2. Print a welcome message: outputs a simple greeting.
3. Determine and print the number of students: user len() to find number.
4. Find and print highest and lowest grades: uses max() and min() to determine the highest and lowest values in student_grades.
6. Print reverse grades: uses reversed() to reverse the sorted list and converts to list.
7. Generate and print a range of grade indices: uses range() to create a list of indices from 1 to number of students
8. Stop

program:

```
def analyze_student_grades():
# sample data
student_names = ["Alice", "Bob", "Charlie", "Diana"]
Student_grades = [85, 92, 78, 90].
# 1. print a welcome message.
print("Welcome to student grades Analyzer \n")
Print("Number of students:", num_students)
# 2. Determine and print number of students.
Print("Welcome to student grades Analyzer \n")
# 3. print type of student names list and grades list
print("\n Type of student_name list:", type(student_names))
print("Type of student_grades list:", type(student_grades))
```

Output:

Welcome to student grades Analyzer!

Number of students: 4

Type of student_names list: <class 'list'>

Type of student_grade list: <class 'list'>

Highest grade: 92

Lowest grade: 78.

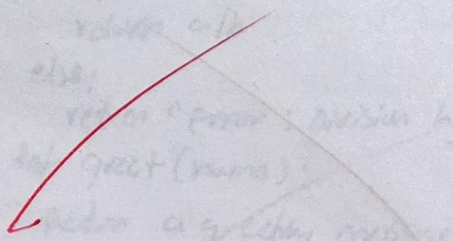
Sorted grades: [78, 85, 90, 92]

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

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

```
# 4. Find print the highest and lowest grade.
highest_grade = max(student_grades)
lowest_grade = min(student_grades)
print("\n 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("\n sorted grades:", sorted_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 a range grade indices from 1 to
number of students
grade_indices = list(range(1, num_students+1))
print("\n grade indices from 1 to number of student:",
grades_indices).
# Run the analysis
analyze_student_grades()
```



output:-

Arithmetic operations:

Sum of 10 and 5 : 15

Difference blw 10 and 5 : 5

product of 10 and 5 : 50.

Quotient of 10 and 5 : 2.0.

Greeting:

Hello, Alice! welcome to program.

Task (6.2): Algorithm

1/9/25

Algorithm:

1. Start the program.
2. User input for number. The program prompts the user to choose an arithmetic operation (addition, subtraction, multiplication, division).
3. Display operation: Based on user's choice, the program performs the choose arithmetic operation defined functions.
4. Display Result: The program displays the results of operation.
5. Stop.

Program:

```
def add(a,b):
    """ Return the sum of two numbers."""
    return a+b.
def subtract(a,b):
    """ return the difference btw two numbers,"""
    return a-b.
def multiply(a,b):
    """ return the quotient product of two numbers,"""
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 user,"""
    return f"Hello, {name}! welcome to program."
def main().
# Demonstrating the use of user-defined functions.
# Arithmetic operation.
num1=10
num2=5.
```

```
print("Arithmetic operations:")
print(f"Arithmetic sum of {num1} & {num2} :", add(num1, num2))
print(f"Difference blw {num1} and {num2} :", substract(num1, num2))
print(f"product of {num1} & {num2} :", multiply(num1, num2))
print(f"quotient of {num1} & {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()
```

Result:

Thus, the python ~~prg~~ progm, using 'Functions' concepts was successfully executed and the output was verified.

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