

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

Date: 03/09/25

6.1 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 `print()`, `len()`, `type()`, `max()`, `min()`, `sorted()`, `reversed()`, and `range()`.

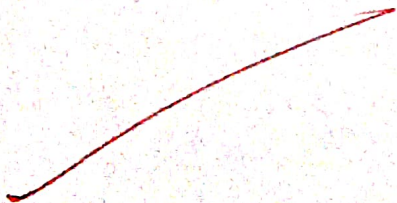
Algorithm:-

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_grade` lists.
5. Find and print highest and lowest uses `max()` and `min()` to determine the highest and lowest values in `student_grade`.
6. print sorted list of grades: uses `sorted()` to sort the grades.
7. print reversed list of grades: uses `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("\nType 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 grades: [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
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)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	5
RECORD (5)	
TOTAL (20)	15
SIGNATURE DATE	

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