

## Task No: 7. Utilizing 'Functions' concepts in Python Programming.

### Aim:

To write the python program using 'Functions' concepts in Python Programming

**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(), max(), min(), sorted(), reversed(), and range().**

### Algorithm:

1. Start the program
2. Print a welcome message: 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 grades: 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: Uses reversed() to reverse the sorted list and converts it to a list.
8. Generate and print a range of grade indices: Uses 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]

    # 1. Print a welcome message
    print("Welcome to the Student Grades Analyzer!\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 grades 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 grade
highest_grade = max(student_grades)
lowest_grade = min(student_grades)
print("\nHighest grade:", highest_grade)
print("Lowest grade:", lowest_grade)

# 5. Print the list of grades sorted in ascending order
sorted_grades = sorted(student_grades)
print("\nSorted 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 of grade indices from 1 to the number of students
grade_indices = list(range(1, num_students + 1))
print("\nGrade indices from 1 to number of students:", grade_indices)

# Run the analysis
analyze_student_grades()

```

**Output:**

```

=====
RESTART: C:/Users/91979/Desktop/print1/t3.py ===
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]

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

```

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 Operation: The program prompts the user to choose an arithmetic operation (addition, subtraction, multiplication, division).
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 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("\nGreeting:")
    print(greet(user_name))

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

```

**Output:**

```
===== RESTART: C:/Users/91979/Desktop/print1/t3.py
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.
```

**Result:**

Thus, the python program using 'Functions' concepts was successfully executed and the output was verified..