

## TASK-7: Utilizing Functions in Python Programming

Aim: TO write the program using 'functions' concepts in Python programming

I.I you are developing a small Python script to analyze and manipulate a list of student grades for a class passed write a python program that satisfies the above requirement using the built in functions `pointc`, `len()`, `typec`, `maxc`, `minc`, `sortedc`, `reversedc` and `rangec`.

### 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\_name list
- 4) Print the type of lists: uses `typec` to show the type of the student\_names and student\_grade lists
- 5) Find and print highest and lowest grades uses `maxc` and `minc` to determine the highest and lowest values in student\_grades
- 6) Print sorted list of grades: uses `sortedc` to sort the grades
- 7) Print reversed list of grades: used `reversedc` the sorted list and converts it to a list
- 8) Generate or print a range of grade 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!")

# 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 name list

and the grades list.

print("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("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("Sorted Grades:", sorted\_grades)

# 6. print the list of grades in reverse order

reversed\_grades = list(reversed(sorted\_grades))

print("Reversed Grades:", reversed\_grades)

(sorted\_grades))

print("Reversed Grades:", reversed\_grades)

7. Generate and print a range of grade indices from 1 to the numbers of students  
grade\\_indices = list(range(1, num\_student))  
print("In Grade indices from 1 to number of student : ", grade\_indices)

Run the analysis  
analyze\_student\_grades()

2 you are tasked with creating a small calculator application to help users performs basic arithmetic operations greet them with a personalized message.

- 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 (addition, subtraction, multiplication, division)
  - 4) Perform operation: Based on the user's choice the program performs the chosen arithmetic operation using the define functions
  - 5) Display result: the program displays the result of the operation
  - 6) STOP

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 grade: [78, 85, 90, 92]

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

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

## Program:

```
def add(a,b):  
    def add return the sum of the numbers (a,b)  
    return a+b  
def subtract(a,b):  
    def subtract return the sum of the numbers (a,b)  
    return a-b  
def multiply(a,b):  
    def multiply return the sum of the numbers (a,b)  
    return a*b  
def divide(a,b):  
    def divide return the sum of the numbers Handle  
    division by zero (a,b)  
    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)
```

output:

(30) Output of the source code is

: (d, 0) 1100

Arithmetic operations

sum of £10 and 5.15

difference between 10 and 5

product of £10 and 5.50

quotient of 10 and 5.20

Greeting

Hello, Alice! welcome to the program

```

point ("Product of num1 and num2:")
    multiply (num1, num2))
point ("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.	
PERFORMANCE (5)	8
RESULT AND ANALYSIS (5)	9
VIVA VOCE (5)	9
RECORD (5)	7
TOTAL (20)	78/100
SIGN WITH DATE	12/12/2018

Result:-

Thus, the Python program using 'functions' concepts was successfully executed and the output was verified.