

# TASK 6. UTILIZING 'FUNCTIONS' CONCEPT IN PYTHON

10/01/25

Programming

Aim :- To write the Python program using 'FUNCTIONS' concepts in Python programming.

Algorithm :-

- 1) Start the program
- 2) Print a welcome message : outputs a sample greeting
- 3) Determine and print the number of students
- 4) Print the type of lists : uses type() to show the type of the student names and student grades lists.
- 5) Print sorted list of grades : uses sorted() to sort the grades
- 6) Generate and print a range indices : uses range() to create a list of index from 1 to the number of students
- 7) Stop

Program :

```
def analyze_student_grades():  
    # Sample data
```

```
student_names = ["Alice", "Bob", "Charlie", "David"]  
student_grades = [85, 92, 78, 90]
```

## output

# Welcome to the student Analyzer!

Number of students: 4

TYPE OF Student\_names list : <Class'list'>

Type of student - grade list : 2 class list ?

Highest grade: 92% Middle D

lonest grade : 7<sup>th</sup> parissaar गवाली

Sco sorted grade: [78, 85, 90, 92]

reversed order: [92, 90, 85, 98] (H)

Grade index from 1 to number of student  
: [1, 2, 3, 97]

#1. print a welcome message

```
Print("Welcome to the Student Grades  
Analyzer!\\n")
```

#2. determine and print the number of  
num-student = len [student-name]

```
Print["Number of students:", num-student]
```

#3. print the type of the student names  
and the grades list

```
Print["In type student-names list:", type  
[student-name])
```

```
Print["Type of student-grade list:", type  
[student-grade])
```

#4. find and print the highest & lowest grade

```
highest-grade = max [student-grade]
```

```
lowest-grade = min [student-grade]
```

```
Print["In highest grade:", highest-grade)
```

```
Print["Lowest grade:", lowest-grade)
```

#5 print the list of grades in reverse order

```
reversed-grades = list [reversed . [sorted-grades]]
```

```
Print["Reversed grades:", reversed-grades]
```

#6 Print the list of grades in reverse order

reversed\_grades = list [reversed [sorted

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,

print ("1 n Grade indices from 1 to number  
of students : ",

# Run the analysis

analyze\_student\_grades()

Grade Indices

✓

Result:- Thus, the program has been verified  
and executed successfully

## Ques Ques 6.2

Aim :- you are tasked with creating a small, calculator application to help users.

Algorithm

- 1) Start the program
- 2) User input for numbers : The Program prompts the user to enter two numbers.
- 3) User input for operation
- 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

```
def add [a,b];  
    """Return the sum of two numbers.""""  
    return a+b  
def subtract [a,b]:  
    """Return the difference b/w two numbers.""""  
    return a-b  
def divide (a,b)  
    """Return the quotient of two numbers.
```

## Output

### Arithmetic Operations

Sum of 10 and 5:15

Difference of 10 and 5:15

Product of 10 and 5:50

Quotient of 10 and 5:20

Greeting:

Hello! Alice; welcome to the program

```

if b1 == 0
    return "b1"
else:
    return "Error: division by zero"

def greet [name]:
    """Return a greeting message for the user."""
    return f"Hello, {name}! Welcome to the program"

# Arithmetic operations
num1 = 10
num2 = 5

print ("Arithmetic operations:")
print (f"Sum of {num1} and {num2}: ", add (num1, num2))
print (f"Difference b/w {num1} and {num2}: ", subtract (num1, num2))
print (f"Quotient of {num1} and {num2}: ", divide (num1, num2))

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

Result: Thus, the python program using
functions' concept was successfully executed
and output is verified

```

| VELTECH                 |    |
|-------------------------|----|
| X No.                   | 10 |
| PERFORMANCE (5)         | 5  |
| RESULT AND ANALYSIS (5) | 5  |
| VIVA VOCE (5)           | 5  |
| RECORD (5)              | 5  |
| TOTAL (20)              | 15 |
| SIGN WITH DATE          | 15 |