

Task no: 7 Utilizing 'Functions' concepts in Python programming

Aim:- To write the Python using 'Functions' concepts in Python

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 requirement 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-name 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 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 : user reversed() the sorted list and converts it to a list~~
- 8) Generate and Print a range of grade indicate from 1 to the number of students

Program :-

```
def analyze - Student - grade ():
```

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 name list
and the grades list

```
print("\n 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("\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)
```

## 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 grade : [78, 85, 90, 92]

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

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

ok

#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 - student + 1))  
Print ("Grade indices from 1 to number of student:", grade - indices)

# Run the analysis

Analyze - Student - grades()

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 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 defined functions

5) Display result: - The program displays the result of the operation

6) Stop

## 1.2 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))
```

# 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:- 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

```

print("Product of {} and {} is: ".format(num1, num2))
print("Quotient of {} and {} is: ".format(num1, num2))

Greeting the user
user_name = "Alice"
print("\n Greeting:")
print(greet(user_name))

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

```

| VEL TECH                |          |
|-------------------------|----------|
| EX No.                  | 7        |
| PERFORMANCE (5)         | 5        |
| RESULT AND ANALYSIS (5) | 5        |
| VIVA VOCE (5)           | 5        |
| RECORD (5)              | 5        |
| TOTAL (20)              | 15       |
| SIGN WITH DATE          | 20/10/19 |

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