

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

Aim:-

To write the python program 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 project write a python program that satisfies the above requirements using the built-in function `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 list
5. print sorted list of grades: Uses `sorted()` to sort the grades.
6. print reversed list of grades: Uses `reversed()` to reverse the sorted list and converts it to a list
7. 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 ("In 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 grade
```

```
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:", Sorted - grades)
```

```
# 6- print the list of grades in reverse  
order
```

```
reversed - grades = list (reversed (Sorted  
- grades))
```

```
print ("Reversed grades:", reversed -  
grades)
```

#1 7. Generate and print a range of grade
indices from 1 to the number of
students

grade - indices - list(range (1, sum-stud-
ents +1))

print ("In Grade indices from 1 to
number of students:", grade-
indices)

Run the analysis

analyze - student - grades()

out put:-

welcome to the student Grades Analy²⁰²¹

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
4. Perform operation: Based on the user's choice, the program performs the chosen 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"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()
```

VELTECH	
No.	
PERFORMANCE (5)	7
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	5
RECORD (5)	5
TOTAL (20)	27
SIGN WITH DATE	

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

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