

Task no:-7 Utilizing 'Functions' concepts in Python Programming

Aim:- To write the Python using 'functions' concepts in Python.

7.7:- 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()` 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()` the sorted list and converts it to a list.
- 8) Generate and print a range of grade ins 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 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
    # grades
    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 7 to the number of students
    grade_indices = list(range(1, num_students + 1))
```

(1) `shorP - fAstuTe` `sortIndex 1st`
`shorP 0th` `studenT 1st`
shorP `shorP` `doA` `{with A}` `- 2nd` `shorP - fAstuTe`
`COPy, CP, CP, CP` `- 2nd` `shorP - fAstuTe`

output: `a pRogram` `encloses` `a fAile` `+ fT`

`welcomE` `to the student` `Grades Analyzer!`
`Number of students: 4`

`TYPE of student - names list; & class list's`

`TYPE of student - grades list; & class list's`

`Highest grade : 92 in shorP. out has`

`out"; lowest grade: 78. To add a/fair`

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

`reversed grade : [92, 90, 85, 78]`

`final Grade indices from 1 to number of student`
`shorP - 2nd` `[1, 2, 3, 4]`

`(shorP - 2nd) 1st = shorP - 1st`

`(shorP - 2nd) 2nd = shorP - 2nd`

`(shorP - 2nd) 3rd = shorP - 3rd`

`(shorP - 2nd) 4th = shorP - 4th`

`(shorP - 2nd) 5th = shorP - 5th`

`(shorP - 2nd) 6th = shorP - 6th`

`(shorP - 2nd) 7th = shorP - 7th`

`(shorP - 2nd) 8th = shorP - 8th`

`(shorP - 2nd) 9th = shorP - 9th`

`(shorP - 2nd) 10th = shorP - 10th`

`(shorP - 2nd) 11th = shorP - 11th`

`(shorP - 2nd) 12th = shorP - 12th`

from 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 calculate 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

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 Progra

def main():
    # Demonstrating the use of user-defined
    # functions
    # Arithmetic operations
    num1=10
    num2=5
```

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

Greeting:

Hello, Alice! welcome to the program

```

Print("Difference between {num1} and {num2};",
Subtract(num1, num2)
Print("Product of {num1} and {num2};",
multiply(num1, num2))
Print("Quotient of {num1} and {num2};", divide
(num1, num2))

# Greeting the user
user_name = "Alice"
Print("\n greeting:")
Print(greet(user_name))

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

```

VELTECH	
EX No.	8
PERFORMANCE (5)	F
RESULT AND ANALYSIS (5)	F
VIVA VOCE (5)	F
RECORD (5)	F
TOTAL (20)	22
SIGN WITH DATE	

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