10/9/24

Task: Utilizing 'Functions' concepts in python Programming

Aim: To write the python program using 'functions' concepts in python Programming

8.1 you are developing a small python script to analyze eand manipulate a list of student grades for a class project. Write a python program that satisfies the above requirements using the bulit-in functins print(), len() type(), max(), min(), sorted(), reversed() and range().

Algorithm:

1) start the program
2) Print a welcome message: output a simple greeting.
3) Determine and print the number of students: Use len() to find number of elements in the student_name list
4) print the type of lists: Use type() to show the type of the student_names and student_grades lists.
5) Find and print highest and lowest grades: Use max() and min() to determine the highest and lowest values in student_grades.
6) print sorted list of grades: Use sorted() to sorted the grades.
7) print reversed list of grades: Use reversed() to reverse the sorted list and convert it to a list.
8) Generate and print a range of grade indices: Use range() to create a list of 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!\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 student names list and the grades list
Print("In type of student name 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)
lower_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)

#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_indicates = list(range(1, num_students+1))
Print("\n Grade indices from 1 to number of students:", grade_indices)

# Run the analysis
analyze_student_grades()

Output:
Welcome to student Analyzer

Number of student : 4

Types of student - name list : <class 'list'>

Types of student - grades list : <class 'list'>

Highest grade : 92

lowest grade : 78

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

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

Grade values from 1 to number of student : [1, 2, 3, 4]

Q.2 You are tasked with creating a small calculator application to help users perform basic arithmetic operations and greets 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 (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.

Program:

```
def add(a,b):
    """Return the sum of two numbers"""
    return a+b
def subtract(a,b):
    """ Return the difference between twonumber"""
    return a-b
def multiply(a,b):
    """ Return the product of twonumbers"""
def divide(a,b):
    """ Return the quotient of two number.
        Handles division by zero"""
    b != 0;
```

Arthemtic Operation:

Sum of 10 and 5: 15

Difference b/w 10 & 5: 5

Product of 10 & 5: 50

~~Sub~~ Quotient of 10 and 5 : 2.0

Greeting

Hello, Slice! welcome to the program.

```
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 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"Product of {num1} and {num2}:", multiply(num1,
                                                    num2))
    print(f"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()
```

VEL TECH	
EX NO.	
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	5
RECORD (5)	5
TOTAL (30)	20

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