Week 7 Assignment

Name : Drishti Durgesh Telgu
Student ID: SM20240093
Unit: ICT_102
Professor: Dr. Duy Nguyen

Tutorial :

This exercise is designed to assess your ability to use modules and clients, handle positional and keyword arguments, write docstrings, work with dictionaries and nested lists, develop a calculator program, and interact with an API.

Part 1: Module and Client

Create a Python module named calculator.py that contains various functions to perform arithmetic operations.

add: Adds two numbers.

subtract: Subtracts the second number from the first.

multiply: Multiplies two numbers.

divide: Divides the first number by the second, includes exception handling for division by zero.

Each function should accept both positional and keyword arguments and include docstrings.

Part 2: Using the Module and Positional/Keyword Arguments

Create a client script named client.py that imports and uses the functions from calculator.py. Demonstrate the use of both positional and keyword arguments.

Part 3: Using Dictionary and Nested List

Extend the calculator.py module to include a function that processes a nested list and returns a dictionary with results of various arithmetic operations.
process_operations: Takes a nested list of operations and returns a dictionary with the results.

Part 4: Developing a Calculator Program
Create an interactive calculator program in client.py that uses the functions from calculator.py.

Part 5: API Integration
Integrate an API into your calculator program. For this exercise, we'll use a simple API to get exchange rates and convert currencies.
get_exchange_rate: Fetches the exchange rate for a given currency pair from a free API.
Update client.py to include currency conversion functionality.

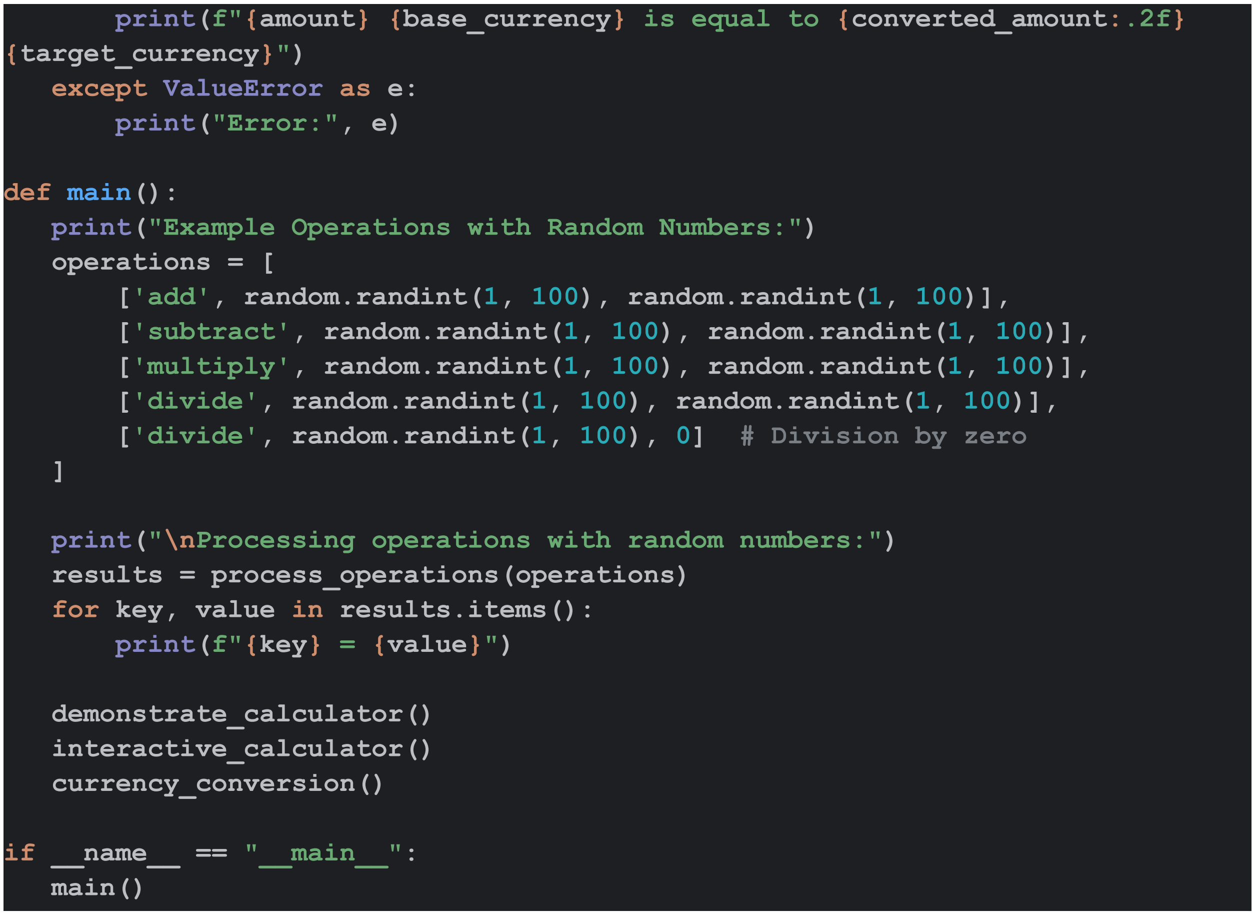
Submission
Save your calculator.py and client.py files.
Ensure all functions are well-documented with comments explaining their purpose.
Test all functions to ensure they work as expected.
Copy and paste the source code to this file, and save it as a PDF.

Program :
Calculator.py

```
def add(a, b):
   return a + b

def subtract(a, b):
   return a - b

def multiply(a, b):
   return a * b

def divide(a, b):
   if b == 0:
       raise ValueError("Cannot divide by zero.")
   return a / b

def process_operations(nested_list):
   results = {}
   for operation in nested_list:
       op, a, b = operation
       if op == 'add':
           results[f"{a} + {b}"] = add(a, b)
       elif op == 'subtract':
           results[f"{a} - {b}"] = subtract(a, b)
       elif op == 'multiply':
           results[f"{a} * {b}"] = multiply(a, b)
       elif op == 'divide':
           try:
               results[f"{a} / {b}"] = divide(a, b)
           except ValueError as e:
               results[f"{a} / {b}"] = str(e)
       else:
           results[f"{a}{op}{b}"] = "Invalid operation"
   return results

def get_exchange_rate(base_currency, target_currency):
   rates = {
       ('USD', 'EUR'): 0.85,
       ('EUR', 'USD'): 1.18,
       ('USD', 'GBP'): 0.75,
       ('GBP', 'USD'): 1.33
   }
   return rates.get((base_currency, target_currency), 1.0)
```

Client.py

```
import random
from calculator import add, subtract, multiply, divide, process_operations,
get_exchange_rate

def demonstrate_calculator():
    print("Positional Arguments:")
    print("Add:", add(10, 5))
    print("Subtract:", subtract(10, 5))
    print("Multiply:", multiply(10, 5))

    print("\nKeyword Arguments:")
    print("Add:", add(a=7, b=3))
    print("Subtract:", subtract(a=20, b=4))
    print("Multiply:", multiply(a=8, b=2))

    try:
        print("\nDivide:")
        print("Divide (positional):", divide(10, 2))
        print("Divide (keyword):", divide(a=10, b=0))
    except ValueError as e:
        print("Error:", e)

def interactive_calculator():
    print("Interactive Calculator")
    while True:
        operation = input("Enter operation (add, subtract, multiply, divide) or
'quit' to exit: ").strip().lower()

        if operation == 'quit':
            break

        try:
            a = float(input("Enter the first number: "))
            b = float(input("Enter the second number: "))

            if operation == 'add':
                print("Result:", add(a, b))
            elif operation == 'subtract':
                print("Result:", subtract(a, b))
            elif operation == 'multiply':
                print("Result:", multiply(a, b))
            elif operation == 'divide':
                print("Result:", divide(a, b))
            else:
                print("Invalid operation")
        except ValueError as e:
            print("Error:", e)

def currency_conversion():
    print("\nCurrency Conversion")
    base_currency = input("Enter base currency code (e.g., USD):
").strip().upper()
    target_currency = input("Enter target currency code (e.g., EUR):
").strip().upper()

    try:
        rate = get_exchange_rate(base_currency, target_currency)
        amount = float(input(f"Enter amount in {base_currency}: "))
        converted_amount = amount * rate
```



```
        print(f"{amount} {base_currency} is equal to {converted_amount:.2f}
{target_currency}")
    except ValueError as e:
        print("Error:", e)

def main():
    print("Example Operations with Random Numbers:")
    operations = [
        ['add', random.randint(1, 100), random.randint(1, 100)],
        ['subtract', random.randint(1, 100), random.randint(1, 100)],
        ['multiply', random.randint(1, 100), random.randint(1, 100)],
        ['divide', random.randint(1, 100), random.randint(1, 100)],
        ['divide', random.randint(1, 100), 0]  # Division by zero
    ]

    print("\nProcessing operations with random numbers:")
    results = process_operations(operations)
    for key, value in results.items():
        print(f"{key} = {value}")

    demonstrate_calculator()
    interactive_calculator()
    currency_conversion()

if __name__ == "__main__":
    main()
```

Output :

```
C:\Users\61411\PycharmProjects\pythonProject\.venv\Scri
pts\python.exe
C:\Users\61411\PycharmProjects\pythonProject\.venv\Lib\
client.py
Example Operations with Random Numbers:

Processing operations with random numbers:
72 + 51 = 123
32 - 99 = -67
14 * 96 = 1344
72 / 42 = 1.7142857142857142
54 / 0 = Cannot divide by zero.
Positional Arguments:
Add: 15
Subtract: 5
```

```
Multiply: 50

Keyword Arguments:
Add: 10
Subtract: 16
Multiply: 16

Divide:
Divide (positional): 5.0
Error: Cannot divide by zero.
Interactive Calculator
Enter operation (add, subtract, multiply, divide) or
'quit' to exit: divide
Enter the first number: 64
Enter the second number: 96
Result: 0.6666666666666666
Enter operation (add, subtract, multiply, divide) or
'quit' to exit:
```