1. Create a client-server program where the client sends two numbers and an operation (+, −, ×, ÷) and the server saves the details in a **JSON or Text file**, giving the answers back to the client.

Server.py

```python
import socket
import json
import os
FILE_NAME = "calculations.json"

if not os.path.exists(FILE_NAME):
    with open(FILE_NAME, "w") as f:
        json.dump([], f)

def save_to_json(data):
    with open(FILE_NAME, "r") as f:
        history = json.load(f)
    history.append(data)
    with open(FILE_NAME, "w") as f:
        json.dump(history, f, indent=4)
```

```python
def calculate(num1, num2, operation):
    try:
        if operation == '+':
            return num1 + num2
        elif operation == '-':
            return num1 - num2
        elif operation == '*':
            return num1 * num2
        elif operation == '/':
            return num1 / num2 if num2 != 0 else "Error: Division by zero"
        else:
            return "Error: Invalid operation"
    except Exception as e:
        return f"Error: {e}"
```

```python
def start_server():
    server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    server_socket.bind(("127.0.0.1", 5555))
    server_socket.listen(5)

    print("Server started. Waiting for clients...")

    while True:
        client_socket, address = server_socket.accept()
        print(f"Connection from {address}")
        data = client_socket.recv(1024).decode()
        if not data:
            client_socket.close()
            continue

        try:
            request = json.loads(data)
            num1 = request["num1"]
            num2 = request["num2"]
            operation = request["operation"]
            result = calculate(num1, num2, operation)
            save_to_json({
                "num1": num1,
                "num2": num2,
                "operation": operation,
                "result": result
            })
```

```python
            response = json.dumps({"result": result})
            client_socket.send(response.encode())

        except Exception as e:
            client_socket.send(json.dumps({"error": str(e)}).encode())

        client_socket.close()

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

Client.py

```python
import socket
import json

def start_client():
    num1 = float(input("Enter first number: "))
    num2 = float(input("Enter second number: "))
    operation = input("Enter operation (+, -, *, /): ")
    request = {
        "num1": num1,
        "num2": num2,
        "operation": operation
    }
    client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    client_socket.connect(("127.0.0.1", 5555))

    client_socket.send(json.dumps(request).encode())

    response = client_socket.recv(1024).decode()
    result = json.loads(response).get("result", "Error")

    print(f"Result: {result}")

    client_socket.close()

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

calculations.json

```json
[
    {
        "num1": 12.0,
        "num2": 12.0,
        "operation": "+",
        "result": 24.0
    }
]
```

```
PS C:\Users\Kainat\Downloads> python CNLAB3TASK1client.py
Enter first number: 12
Enter second number: 12
Enter operation (+, -, *, /): +
Result: 24.0
```

```
PS C:\Users\Kainat\Downloads> python CNLAB3TASK1server.py
Server started. Waiting for clients...
Connection from ('127.0.0.1', 59319)
```

2. Create a client-server program where the client requests the grading scheme (enter grade points) and the server responds with the answer according to the given grading scheme (respond letter grade and qualification according to the client's entered marks).

Letter grade	Grade point	Qualification
A +	4.33	Excellent
A	4.00	Excellent
A –	3.66	Very good
B +	3.33	Very good
B	3.00	Very good
B –	2.66	Good
C +	2.33	Good
C	2.00	Good
C –	1.66	Passable
D +	1.33	Passable
D	1.00	Passable
E	0.00	Failure

Server.py

```python
import socket
import json

grading_scheme = [
    (4.33, "A+", "Excellent"),
    (4.00, "A", "Excellent"),
    (3.66, "A-", "Very good"),
    (3.33, "B+", "Very good"),
    (3.00, "B", "Very good"),
    (2.66, "B-", "Good"),
    (2.33, "C+", "Good"),
    (2.00, "C", "Good"),
    (1.66, "C-", "Passable"),
    (1.33, "D+", "Passable"),
    (1.00, "D", "Passable"),
    (0.00, "E", "Failure")
]

def get_grade_info(grade_point):
    for gp, grade, qualification in grading_scheme:
        if grade_point >= gp:
            return grade, qualification
    return "E", "Failure"
```

```python
def start_server():
    server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    server_socket.bind(("127.0.0.1", 6000))
    server_socket.listen(5)
    print("Server started. Waiting for clients...")

    while True:
        client_socket, address = server_socket.accept()
        print(f"Connection from {address}")

        try:
            data = client_socket.recv(1024).decode()
            grade_point = float(data)

            letter_grade, qualification = get_grade_info(grade_point)

            response = {
                "letter_grade": letter_grade,
                "qualification": qualification
            }
            client_socket.send(json.dumps(response).encode())
        except Exception as e:
            client_socket.send(json.dumps({"error": str(e)}).encode())

        client_socket.close()

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

client.py

```python
import socket
import json

def start_client():
    grade_point = float(input("Enter your grade point: "))
    client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    client_socket.connect(("127.0.0.1", 6000))

    client_socket.send(str(grade_point).encode())

    response = client_socket.recv(1024).decode()
    result = json.loads(response)

    if "error" in result:
        print(f"Error: {result['error']}")
    else:
        print(f"Letter Grade: {result['letter_grade']}")
        print(f"Qualification: {result['qualification']}")

    client_socket.close()

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

```
PS C:\Users\Kainat\Downloads> python CNLAB3TASK2server.py
Server started. Waiting for clients...
Connection from ('127.0.0.1', 49704)
Connection from ('127.0.0.1', 49707)
```

```
PS C:\Users\Kainat\Downloads> python CNLAB3TASK2client.py
Enter your grade point: 4.00
Letter Grade: A
Qualification: Excellent
```