DAY 25:

ASSIGNMENT 5:

Task 5: Java Networking and Serialization

Develop a basic TCP client and server application where the client sends a serialized object with 2 numbers and operation to be performed on them to the server, and the server computes the result and sends it back to the client. for eg, we could send 2, 2, ""+"" which would mean 2 + 2

ANSWER:

```
import java.io.*;

import java.net.*;


class OperationData implements Serializable {

   private static final long serialVersionUID = 1L;

   private double num1;

   private double num2;

   private String operation;


   public OperationData(double num1, double num2, String operation) {

      this.num1 = num1;

      this.num2 = num2;

      this.operation = operation;

   }


   public double getResult() {

      switch (operation) {

         case "+":

            return num1 + num2;

         case "-":

            return num1 - num2;

         case "*":

            return num1 * num2;
            case "/":
                if (num2 != 0)
                    return num1 / num2;
                else
                    throw new ArithmeticException("Division by zero!");
            default:
                throw new IllegalArgumentException("Invalid operation: " + operation);
        }
    }
}

public class ClientServer {
    public static void main(String[] args) {
        // Start server
        new Thread(() -> Server.start()).start();

        // Delay to ensure the server starts before the client tries to connect
        try {
            Thread.sleep(1000);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }

        // Start client
        Client.start();
    }
}

class Server {
    public static void start() {
        try {
```

```
ServerSocket serverSocket = new ServerSocket(9999);
      System.out.println("Server started...");

      while (true) {
        Socket socket = serverSocket.accept();
        System.out.println("Client connected: " + socket);

        ObjectInputStream objectInputStream = new ObjectInputStream(socket.getInputStream());
        ObjectOutputStream objectOutputStream = new
ObjectOutputStream(socket.getOutputStream());

        // Receive serialized object from client
        Object receivedObject = objectInputStream.readObject();
        if (receivedObject instanceof OperationData) {
          OperationData operationData = (OperationData) receivedObject;
          // Perform the operation
          double result = operationData.getResult();

          // Send the result back to the client
          objectOutputStream.writeDouble(result);
          objectOutputStream.flush();
        }
        // Close connections
        objectInputStream.close();
        objectOutputStream.close();
        socket.close();
      }
    } catch (IOException | ClassNotFoundException e) {
      e.printStackTrace();      }
  }
}
```

```
class Client {
    public static void start() {
        try {
            Socket socket = new Socket("localhost", 9999);
            System.out.println("Connected to server...");

            ObjectOutputStream objectOutputStream = new
ObjectOutputStream(socket.getOutputStream());
            ObjectInputStream objectInputStream = new ObjectInputStream(socket.getInputStream());

            // Create the operation data object
            OperationData operationData = new OperationData(2, 2, "+");

            // Send serialized object to server
            objectOutputStream.writeObject(operationData);
            objectOutputStream.flush();

            // Receive result from server
            double result = objectInputStream.readDouble();
            System.out.println("Result from server: " + result);

            // Close connections
            objectOutputStream.close();
            objectInputStream.close();
            socket.close();
        } catch (IOException e) {
            e.printStackTrace();
}
    }
}
```