## 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

```
package test;
import java.io.*;
import java.net.*;
class CalculationRequest implements Serializable {
private static final long serialVersionUID = 1L;
private int number1;
private int number2;
private String operation;
public CalculationRequest(int number1, int number2, String operation) {
this.number1 = number1;
this.number2 = number2;
this.operation = operation;
}
public int getNumber1() {
return number1;
}
public int getNumber2() {
return number2;
}
public String getOperation() {
return operation;
}
}
public class TCPServer {
public static void main(String[] args) {
int port=5555;
try (ServerSocket serverSocket=new ServerSocket(port)) {
System.out.println("Server is running and listening on port " + port);
```

```
while (true) {
try (Socket socket=serverSocket.accept()) {
System.out.println("Client connected: "+socket);
ObjectInputStream objectInput=new ObjectInputStream(socket.getInputStream());
CalculationRequest request=(CalculationRequest) objectInput.readObject();
System.out.println("Received request: " +
request.getNumber1()+" "+request.getOperation()+" "+request.getNumber2());
int result=calculate(request.getNumber1(), request.getNumber2(), request.getOperation());
ObjectOutputStream objectOutput=new ObjectOutputStream(socket.getOutputStream());
objectOutput.writeInt(result);
objectOutput.flush();
System.out.println("Sent result: " + result);
} catch (IOException | ClassNotFoundException e) {
e.printStackTrace();
}
}
} catch (IOException e) {
e.printStackTrace();
}
}
private static int calculate(int num1, int num2, String operation) {
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 error");
}
default:
throw new IllegalArgumentException("Unsupported operation: " + operation);
}
}
}
```