{··}

Question 5

Write a RMI program to calculate tax depending on the salary.

TaxCalculator.java:

```java
import java.rmi.Remote;
import java.rmi.RemoteException;

public interface TaxCalculator extends Remote {
    double calculateTax(double salary) throws RemoteException;
}
```

TaxCalculatorImpl.java:

```java
import java.rmi.RemoteException;
import java.rmi.server.UnicastRemoteObject;

public class TaxCalculatorImpl extends UnicastRemoteObject implements TaxCalculator {

    public TaxCalculatorImpl() throws RemoteException {
        super();
    }

    public double calculateTax(double salary) throws RemoteException {
        double tax;

        if (salary <= 10000) {
            tax = 0;
        } else if (salary <= 25000) {
            tax = 0.025 * salary;
        } else if (salary <= 100000) {
            tax = 0.05 * salary;
        } else if (salary <= 1000000) {
            tax = 0.075 * salary;
        } else {
            tax = 0.1 * salary;
        }

        return tax;
    }
}
```

Server.java:

```java
import java.rmi.Naming;
import java.rmi.registry.LocateRegistry;

public class Server {
    public static void main(String args[]) {
        try {
            TaxCalculatorImpl calculator = new TaxCalculatorImpl();

            //Create and start the RMI registry on port 1099
            LocateRegistry.createRegistry(1099);

            //Bind the remote object's stub in the registry
            Naming.rebind("rmi://localhost/TaxCalculator", calculator);
            System.out.println("Server started");
        } catch (Exception e) {
            System.err.println("Server exception: " + e.toString());
            e.printStackTrace();
        }
    }
}
```

Client.java:

```java
import java.rmi.Naming;

public class Client {
    public static void main(String args[]) {
        try {
            TaxCalculator calculator = (TaxCalculator) Naming.lookup("rmi://localhost/TaxCalculator");

            double salary = 50000;

            double tax = calculator.calculateTax(salary);

            System.out.println("Tax: " + tax);
        }
        catch (Exception e) {
            System.err.println("Client exception: " + e.toString());
            e.printStackTrace();
        }
    }
}
```

Output:

Compile all the files, then:

1. **Run *Server* file first:**

`Server started`

2. **Then, open a new tab in the same path where your java files are, then run *Client* file:**

`Tax: 2500.0`