**Client.java**

|  |
| --- |
| import java.rmi.registry.LocateRegistry;  import java.rmi.registry.Registry;  import java.util.\*  public class Client {  public static void main(String[] args) {  Scanner sc = new Scanner(System.in);  try {  Registry registry = LocateRegistry.getRegistry("localhost", 1099);  RemoteInterface remoteObj = (RemoteInterface) registry.lookup("CalculatorService");  int a = sc.nextInt();  int b = sc.nextInt();  String name = sc.next();  System.out.println("Sum: " + remoteObj.add(a, b));  System.out.println("Difference: " + remoteObj.subtract(a, b));  System.out.println("Product: " + remoteObj.multiply(a, b));  System.out.println("Quotient: " + remoteObj.divide(a, b));  System.out.println("Enter Student name");  System.out.println("Information\n" + remoteObj.info(name));  } catch (Exception e) {  System.err.println("Client exception: " + e.toString());  e.printStackTrace();  }  }  } |

**Server.java**

|  |
| --- |
| import java.rmi.registry.LocateRegistry;  import java.rmi.registry.Registry;  public class Server {  public static void main(String[] args) {  try {  RemoteInterface remoteObj = new RemoteImplementation();  Registry registry = LocateRegistry.createRegistry(1099);  registry.rebind("CalculatorService", remoteObj);  System.out.println("Server is running...");  } catch (Exception e) {  System.err.println("Server exception: " + e.toString());  e.printStackTrace();  }  }  } |

**RemoteImplementation.java**

|  |
| --- |
| import java.rmi.RemoteException;  import java.rmi.server.UnicastRemoteObject;  public class RemoteImplementation extends UnicastRemoteObject implements RemoteInterface {  protected RemoteImplementation() throws RemoteException {  super();  }  @Override  public int add(int a, int b) throws RemoteException {  return a + b;  }  @Override  public int subtract(int a, int b) throws RemoteException {  return a - b;  }  @Override  public int multiply(int a, int b) throws RemoteException {  return a \* b;  }  @Override  public int divide(int a, int b) throws RemoteException {  return a / b;  }  @Override  public String info(String name) throws RemoteException {  String name\_info = "Name: Nikhil\nRoll.No: B57\nComputer Science and Engineering";  if(name.equals("Nikhil")){  System.out.println("Name: Nikhil Vilas Jadhav");  System.out.println("Roll.No: B57");  System.out.println("Computer Science and Engineering");  }else{  System.out.println("Data is not found");  }  return name\_info;  }  } |

|  |
| --- |
| import java.rmi.Remote;  import java.rmi.RemoteException;  public interface RemoteInterface extends Remote {  int add(int a, int b) throws RemoteException;  int subtract(int a, int b) throws RemoteException;  int multiply(int a, int b) throws RemoteException;  int divide(int a, int b) throws RemoteException;  String info(String name) throws RemoteException;  } |

|  |
| --- |
| PS C:\Users\nikhi\Desktop\Practical\DC\pract5> java Client  45  25  Nikhil  Sum: 70  Product: 1125  Quotient: 1  Enter Student name  Information  Name: Nikhil  Roll.No: B57  Computer Science and Engineering  PS C:\Users\nikhi\Desktop\Practical\DC\pract5> java Server  Server is running...  Name: Nikhil Vilas Jadhav  Roll.No: B57  Computer Science and Engineering |

**RemoteInterface.java**