

Unit 2

Q1) RPC using Datagram socket: Implement a Date Time server containing date() and time()

DateTimeServer.java:

```
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.text.SimpleDateFormat;
import java.util.Date;
public class DateTimeServer {
    private static final int PORT = 9876;
    public static void main(String[] args) {
        try (DatagramSocket socket = new DatagramSocket(PORT))
        { System.out.println("DateTime Server is running on port " + PORT);
          byte[] receiveData = new byte[1024];
          while (true) {
              DatagramPacket receivePacket = new DatagramPacket(receiveData,
receiveData.length);
              socket.receive(receivePacket);
              String request = new String(receivePacket.getData(), 0,
receivePacket.getLength());
              InetAddress clientAddress = receivePacket.getAddress();
              int clientPort = receivePacket.getPort();
              String response = handleRequest(request.trim());
              byte[] sendData = response.getBytes();
              DatagramPacket sendPacket = new DatagramPacket(sendData,
sendData.length, clientAddress, clientPort);
              socket.send(sendPacket);
          }
        } catch (Exception e)
        { e.printStackTrace();
        }
    }
    private static String handleRequest(String request)
    { switch (request.toLowerCase()) {
        case "date":
            return getDate();
        case "time":
```

```

        return getCurrentTime();
    default:
        return "Unknown command. Use 'date' or 'time'.";
    }
}

private static String getCurrentDate() {
    SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");
    return sdf.format(new Date());
}

private static String getCurrentTime() {
    SimpleDateFormat sdf = new SimpleDateFormat("HH:mm:ss");
    return sdf.format(new Date());
}
}

```

DateTimeClient.java:

```

import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.util.Scanner;

public class DateTimeClient {
    private static final String SERVER_ADDRESS = "localhost";
    private static final int SERVER_PORT = 9876;
    public static void main(String[] args) {
        try (DatagramSocket socket = new DatagramSocket();
            Scanner scanner = new Scanner(System.in))
        {
            InetAddress serverAddress =
                InetAddress.getByName(SERVER_ADDRESS);
            while (true) {
                System.out.println("Enter 'date' to get the current date, 'time' to get the
current time, or 'exit' to quit:");
                String input = scanner.nextLine();
                if (input.equalsIgnoreCase("exit"))
                    {break;}
            }
            byte[] sendData = input.getBytes();

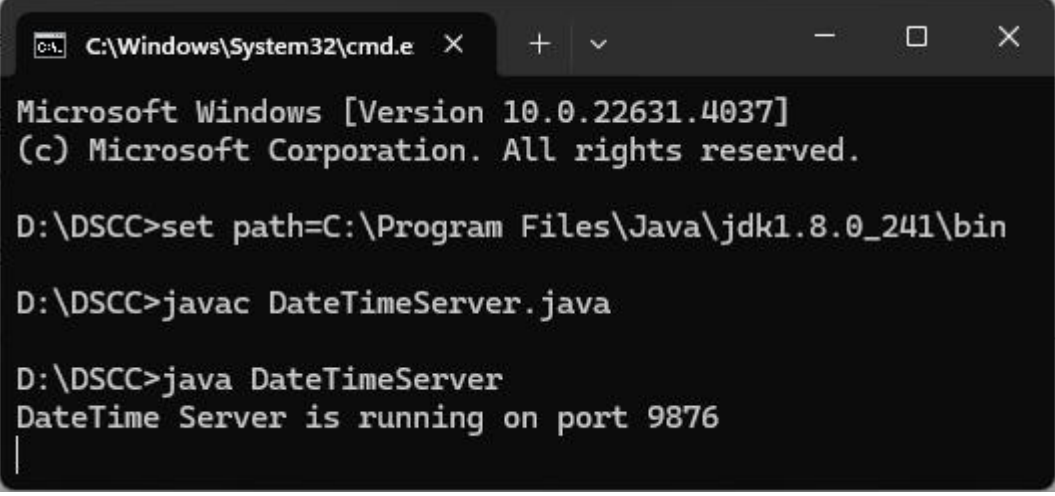
```

```

        DatagramPacket sendPacket = new DatagramPacket(sendData,
sendData.length, serverAddress, SERVER_PORT);
        socket.send(sendPacket);
        byte[] receiveData = new byte[1024];
        DatagramPacket receivePacket = new DatagramPacket(receiveData,
receiveData.length);
        socket.receive(receivePacket);
        String response = new String(receivePacket.getData(), 0,
receivePacket.getLength());
        System.out.println("Response from server: " + response);
    }
} catch (Exception e)
    {e.printStackTrace();
    }
}
}
}

```

Output:



```

C:\Windows\System32\cmd.e
Microsoft Windows [Version 10.0.22631.4037]
(c) Microsoft Corporation. All rights reserved.

D:\DSCC>set path=C:\Program Files\Java\jdk1.8.0_241\bin

D:\DSCC>javac DateTimeServer.java

D:\DSCC>java DateTimeServer
DateTime Server is running on port 9876
|

```

```
C:\Windows\System32\cmd.e  X  +  v  -  □  X
Microsoft Windows [Version 10.0.22631.4037]
(c) Microsoft Corporation. All rights reserved.

D:\DSCC>set path=C:\Program Files\Java\jdk1.8.0_241\bin

D:\DSCC>javac DateTimeClient.java

D:\DSCC>java DateTimeClient
Enter 'date' to get the current date, 'time' to get the current time, or 'exit' to quit:
date
Response from server: 2024-09-13
Enter 'date' to get the current date, 'time' to get the current time, or 'exit' to quit:
time
Response from server: 13:54:46
Enter 'date' to get the current date, 'time' to get the current time, or 'exit' to quit:
exit

D:\DSCC>|
```

Q2) Implement an Age calculator server which displays the age where the client provides his/her birth year

AgeCalculatorServer.java:

```
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.util.Calendar;
public class AgeCalculatorServer
{ private static final int PORT = 9876;
  public static void main(String[] args) {
    try (DatagramSocket socket = new DatagramSocket(PORT))
    { System.out.println("Age Calculator Server is running on port " +
PORT);
      byte[] receiveData = new byte[1024];
      while (true) {
        DatagramPacket receivePacket = new DatagramPacket(receiveData,
receiveData.length);
        socket.receive(receivePacket);
        // Extracting the request data
        String request = new String(receivePacket.getData(), 0,
receivePacket.getLength()).trim();
        InetAddress clientAddress = receivePacket.getAddress();
        int clientPort = receivePacket.getPort();
        // Handle the request and generate a response
        String response = handleRequest(request);
        byte[] sendData = response.getBytes();
        // Send the response back to the client
        DatagramPacket sendPacket = new DatagramPacket(sendData,
sendData.length, clientAddress, clientPort);
        socket.send(sendPacket);
      }
    } catch (Exception e)
    { e.printStackTrace();
    }
  }
  private static String handleRequest(String request)
  { try {
```

```

        int birthYear = Integer.parseInt(request);
        int currentYear = Calendar.getInstance().get(Calendar.YEAR);
        int age = currentYear - birthYear;
        return String.valueOf(age);
    } catch (NumberFormatException e) {
        return "Invalid year format. Please provide a valid year.";
    } catch (Exception e) {
        return "An error occurred: " + e.getMessage();
    }
}
}

```

AgeCalculatorClient.java:

```

import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.util.Scanner;

public class AgeCalculatorClient {
    private static final String SERVER_ADDRESS = "localhost";
    private static final int SERVER_PORT = 9876;
    public static void main(String[] args) {
        try (DatagramSocket socket = new DatagramSocket();
             Scanner scanner = new Scanner(System.in))
        {
            InetAddress serverAddress =
InetAddress.getByAddress(SERVER_ADDRESS);
            while (true) {
                System.out.println("Enter your birth year (or 'exit' to quit):");
                String input = scanner.nextLine();
                if (input.equalsIgnoreCase("exit"))
                    {break;
                }
                byte[] sendData = input.getBytes();
                DatagramPacket sendPacket = new DatagramPacket(sendData,
sendData.length, serverAddress, SERVER_PORT);
                socket.send(sendPacket);
                byte[] receiveData = new byte[1024];

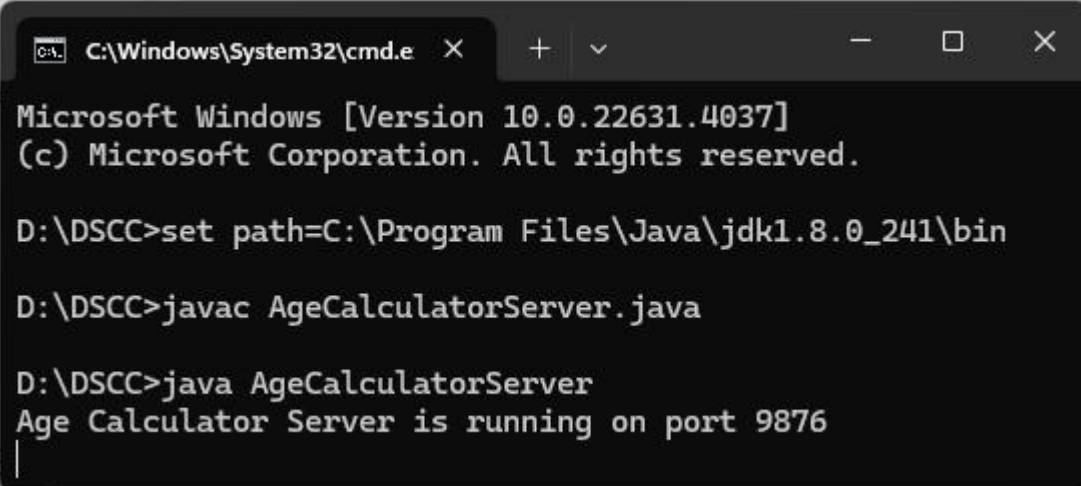
```

```

        DatagramPacket receivePacket = new DatagramPacket(receiveData,
receiveData.length);
        socket.receive(receivePacket);
        String response = new String(receivePacket.getData(), 0,
receivePacket.getLength());
        System.out.println("Your age is: " + response);
    }
} catch (Exception e)
{e.printStackTrace();
}
}
}

```

Output:



```

C:\Windows\System32\cmd.e X + v - □ X
Microsoft Windows [Version 10.0.22631.4037]
(c) Microsoft Corporation. All rights reserved.

D:\DSCC>set path=C:\Program Files\Java\jdk1.8.0_241\bin

D:\DSCC>javac AgeCalculatorServer.java

D:\DSCC>java AgeCalculatorServer
Age Calculator Server is running on port 9876
|

```



```
C:\Windows\System32\cmd.e X + v - □ X
Microsoft Windows [Version 10.0.22631.4037]
(c) Microsoft Corporation. All rights reserved.

D:\DSCC>set path=C:\Program Files\Java\jdk1.8.0_241\bin

D:\DSCC>javac AgeCalculatorClient.java

D:\DSCC>java AgeCalculatorClient
Enter your birth year (or 'exit' to quit):
2002
Your age is: 22
Enter your birth year (or 'exit' to quit):
exit

D:\DSCC>
```

Q3) Implement server which greets client according to the current time of the server “good morning”, “good afternoon”, “Good evening” and “Good Night”

GreetingServer.java:

```
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.util.Calendar;
public class GreetingServer {
    private static final int PORT = 9876;
    public static void main(String[] args) {
        try (DatagramSocket socket = new DatagramSocket(PORT))
        { System.out.println("Greeting Server is running on port " + PORT);
          byte[] receiveData = new byte[1024];
          while (true) {
              DatagramPacket receivePacket = new DatagramPacket(receiveData,
receiveData.length);
              socket.receive(receivePacket);
              // Extracting the client's message (which includes the client's name)
              String request = new String(receivePacket.getData(), 0,
receivePacket.getLength()).trim();
              InetAddress clientAddress = receivePacket.getAddress();
              int clientPort = receivePacket.getPort();
              // Generate the personalized greeting based on the current time
              String response = getGreeting(request);
              byte[] sendData = response.getBytes();
              // Send the personalized greeting back to the client
              DatagramPacket sendPacket = new DatagramPacket(sendData,
sendData.length, clientAddress, clientPort);socket.send(sendPacket);
          }
        } catch (Exception e)
        {e.printStackTrace();
        }
    }
    private static String getGreeting(String name)
    {Calendar now = Calendar.getInstance();
```

```

int hour = now.get(Calendar.HOUR_OF_DAY);
String greeting;
if (hour >= 5 && hour < 12)
    {greeting = "Good morning";
} else if (hour >= 12 && hour < 17)
    {greeting = "Good afternoon";
} else if (hour >= 17 && hour < 21)
    {greeting = "Good evening";
} else {
    greeting = "Good night";
}
return greeting + ", " + name + "!";
}
}

```

GreetingClient.java:

```

import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.util.Scanner;
public class GreetingClient {
    private static final String SERVER_ADDRESS = "localhost";
    private static final int SERVER_PORT = 9876;
    public static void main(String[] args) {
        try (DatagramSocket socket = new DatagramSocket();
            Scanner scanner = new Scanner(System.in))
        { InetAddress serverAddress =
InetAddress.getByName(SERVER_ADDRESS);
            while (true) {
                System.out.println("Enter your name (or 'exit' to quit):");
                String name = scanner.nextLine();
                if (name.equalsIgnoreCase("exit"))
                    {break;
                }
                byte[] sendData = name.getBytes();
                DatagramPacket sendPacket = new DatagramPacket(sendData,
sendData.length, serverAddress, SERVER_PORT);

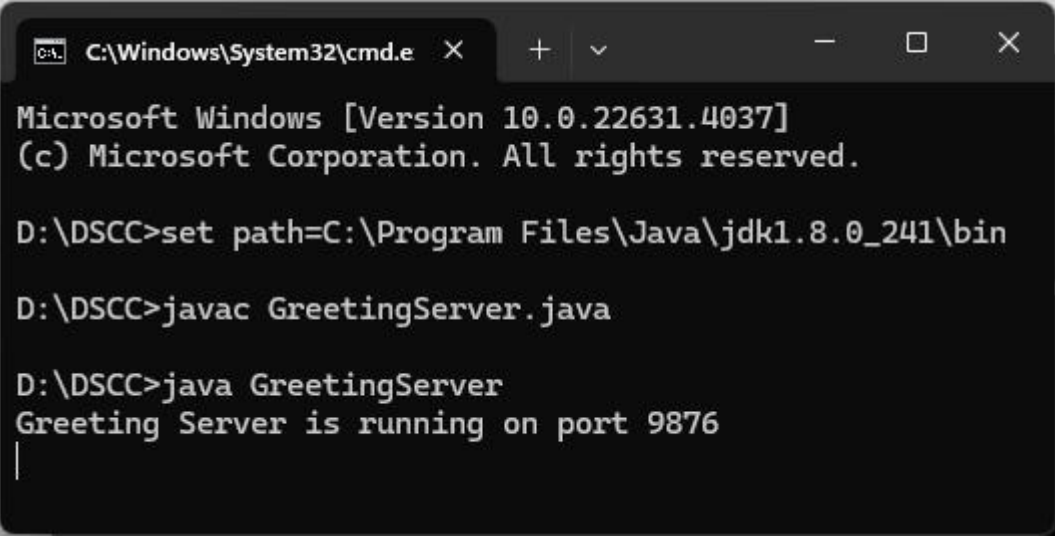
```

```

        socket.send(sendPacket);
        byte[] receiveData = new byte[1024];
        DatagramPacket receivePacket = new DatagramPacket(receiveData,
receiveData.length);
        socket.receive(receivePacket);
        String response = new String(receivePacket.getData(), 0,
receivePacket.getLength());
        System.out.println("Server says: " + response);
    }
} catch (Exception e)
    {e.printStackTrace();
    }
}
}

```

Output:



```

C:\Windows\System32\cmd.e
Microsoft Windows [Version 10.0.22631.4037]
(c) Microsoft Corporation. All rights reserved.

D:\DSCC>set path=C:\Program Files\Java\jdk1.8.0_241\bin

D:\DSCC>javac GreetingServer.java

D:\DSCC>java GreetingServer
Greeting Server is running on port 9876
|

```



C:\Windows\System32\cmd.e



Microsoft Windows [Version 10.0.22631.4037]
(c) Microsoft Corporation. All rights reserved.

D:\New folder (4)>java GreetingClient.java

Enter your name (or 'exit' to quit):

Jitesh Boi

Server says: Good evening, Jitesh Boi!

Enter your name (or 'exit' to quit):

|

Q4) Implement a Server calculator containing ADD(), MUL(), SUB(), etc using datagramsocket

CalculatorServer.java:

```
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
public class CalculatorServer {
    private static final int PORT = 9876;
    public static void main(String[] args) {
        try (DatagramSocket socket = new DatagramSocket(PORT))
        { System.out.println("Calculator Server is running on port " + PORT);
          byte[] receiveData = new byte[1024];
          while (true) {
              DatagramPacket receivePacket = new DatagramPacket(receiveData,
receiveData.length);
              socket.receive(receivePacket);
              // Extracting the client's request
              String request = new String(receivePacket.getData(), 0,
receivePacket.getLength()).trim();
              InetAddress clientAddress = receivePacket.getAddress();
              int clientPort = receivePacket.getPort();
              // Process the request and generate a response
              String response = processRequest(request);
              byte[] sendData = response.getBytes();
              // Send the result back to the client
              DatagramPacket sendPacket = new DatagramPacket(sendData,
sendData.length, clientAddress, clientPort);
              socket.send(sendPacket);
          }
        } catch (Exception e)
        { e.printStackTrace();
        }
    }
    private static String processRequest(String request)
    {try {
        // Split the request into components
        String[] parts = request.split(" ");
```

```

    if (parts.length != 3) {
        return "Invalid request format. Use: OPERATOR operand1 operand2";
    }
    // Parse the operation and operands
    String operator = parts[0];
    double operand1 = Double.parseDouble(parts[1]);
    double operand2 = Double.parseDouble(parts[2]);
    // Perform the operation
    switch (operator.toUpperCase())
    {
        case "ADD":
            return String.valueOf(operand1 + operand2);
        case "SUB":
            return String.valueOf(operand1 - operand2);
        case "MUL":
            return String.valueOf(operand1 * operand2);
        case "DIV":
            if (operand2 == 0) {
                return "Division by zero is not allowed.";
            }
            return String.valueOf(operand1 / operand2);
        default:
            return "Unknown operator. Use ADD, SUB, MUL, or DIV.";
    }
} catch (NumberFormatException e) {
    return "Invalid number format. Ensure operands are valid numbers.";
} catch (Exception e) {
    return "An error occurred: " + e.getMessage();
}
}
}

```

CalculatorClient.java:

```

import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.util.Scanner;

```

```

public class CalculatorClient {
    private static final String SERVER_ADDRESS = "localhost";
    private static final int SERVER_PORT = 9876;
    public static void main(String[] args) {
        try (DatagramSocket socket = new DatagramSocket();
            Scanner scanner = new Scanner(System.in))
        {
            InetAddress serverAddress =
InetAddress.getByName(SERVER_ADDRESS);
            while (true) {
                System.out.println("Enter your request (e.g., ADD 5 3, SUB 10 2, or
'exit' to quit):");
                String request = scanner.nextLine();
                if (request.equalsIgnoreCase("exit"))
                    {break;
                }
                byte[] sendData = request.getBytes();
                DatagramPacket sendPacket = new DatagramPacket(sendData,
sendData.length, serverAddress, SERVER_PORT);
                socket.send(sendPacket);
                byte[] receiveData = new byte[1024];
                DatagramPacket receivePacket = new DatagramPacket(receiveData,
receiveData.length);
                socket.receive(receivePacket);
                String response = new String(receivePacket.getData(), 0,
receivePacket.getLength());
                System.out.println("Server response: " + response);
            }
        } catch (Exception e)
        {
            e.printStackTrace();
        }
    }
}

```


Output:

```
C:\Windows\System32\cmd.e X + - □ X

D:\DSCC>set path=C:\Program Files\Java\jdk1.8.0_241\bin

D:\DSCC>javac CalculatorServer.java

D:\DSCC>java CalculatorServer
Calculator Server is running on port 9876
|
```

```
C:\Windows\System32\cmd.e X + - □ X

Microsoft Windows [Version 10.0.22631.4037]
(c) Microsoft Corporation. All rights reserved.

D:\DSCC>set path=C:\Program Files\Java\jdk1.8.0_241\bin

D:\DSCC>javac CalculatorClient.java

D:\DSCC>java CalculatorClient
Enter your request (e.g., ADD 5 3, SUB 10 2, or 'exit' to quit):
ADD 19 18
Server response: 37.0
Enter your request (e.g., ADD 5 3, SUB 10 2, or 'exit' to quit):
SUB 19 18
Server response: 1.0
Enter your request (e.g., ADD 5 3, SUB 10 2, or 'exit' to quit):
MUL 19 18
Server response: 342.0
Enter your request (e.g., ADD 5 3, SUB 10 2, or 'exit' to quit):
DIV 19 18
Server response: 1.0555555555555556
Enter your request (e.g., ADD 5 3, SUB 10 2, or 'exit' to quit):
exit

D:\DSCC>|
```

Q5) RPC to implement an Equation solver using Datagram. The client should provide an equation to the Server through an interface. The server will solve the expression given by the client. $(a-b)^2 = a^2 - 2ab + b^2$; If $a = 5$ and $b = 2$ then return value = $5^2 - 2 \cdot 5 \cdot 2 + 2^2 = 9$.

intfEqSolve.java:

```
import java.rmi.*;
public interface intfEqSolve extends Remote
{
    public int solveEq1(int a,int b)throws RemoteException;
    public int solveEq2(int a,int b)throws RemoteException;
    public int solveEq3(int a,int b)throws RemoteException;
    public int solveEq4(int a,int b)throws RemoteException;
}
```

implEqSolve.java:

```
import java.rmi.*; import java.rmi.server.*;
public class implEqSolve extends UnicastRemoteObject implements
intfEqSolve
{ public implEqSolve()throws RemoteException {}
    public int solveEq1(int a, int b)throws RemoteException
    { int ans=(a*a)-(2*a*b)+(b*b);
    return ans;
    }
    public int solveEq2(int a,int b)throws RemoteException
    { int ans=(a*a)+(2*a*b)+(b*b);
    return ans;
    }
    public int solveEq3(int a,int b)throws RemoteException
    { int ans=(a*a*a)-(3*a*a*b)+(3*a*b*b)-(b*b*b); return ans ;
    }
    public int solveEq4(int a,int b)throws RemoteException
    { int ans=(a*a*a)+(3*a*a*b)+(3*a*b*b)+(b*b*b); return ans;
    } }
}
```

serverEqSolve.java:

```
import java.io.*;
import java.net.*;
```

```

import java.rmi.*;
public class serverEqSolve {
public static void main(String[]args)
{ try
{
implEqSolve obj=new implEqSolve();
Naming.rebind("hello",obj);
}
catch(Exception e)
{
System.out.println(e);
}}
}

```

clientEqSolve.java:

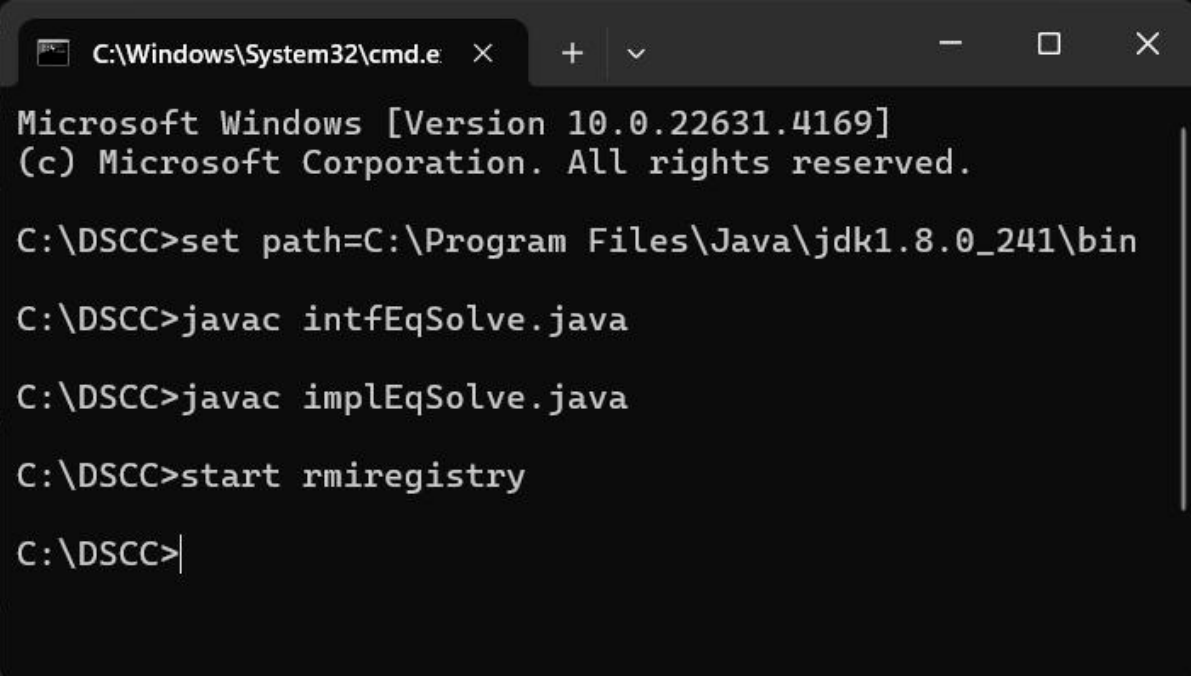
```

import java.io.*;
import java.net.*;
import java.rmi.*;
public class clientEqSolve { public static void main(String[]args)
{ try
{
int num1, num2, res=0, choice;
intfEqSolve object=(intfEqSolve)Naming.lookup("hello");
BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
System.out.println("Equations:-");
System.out.println("1.(a-b)2"); System.out.println("2.(a+b)2");
System.out.println("3.(a-b)3");
System.out.println("4.(a+b)3");
System.out.println("Choose the equation:");
choice=Integer.parseInt(br.readLine());
System.out.println("Enter the value of a and b");
num1=Integer.parseInt(br.readLine()); num2=Integer.parseInt(br.readLine());
switch(choice)
{ case 1:
res=object.solveEq1(num1,num2);
break;
case 2:
res=object.solveEq2(num1,num2);

```

```
break;
case 3:
res=object.solveEq3(num1,num2);
break;
case 4:
res=object.solveEq4(num1,num2);
break;
default:
System.out.println("Invalid option");
break;
}
System.out.println("the answer is"+res);
}
catch(Exception e)
{
System.out.println("Exception:"+e);
} } }
```

Output:



```
C:\Windows\System32\cmd.e  ×  +  ▾  -  □  ×

Microsoft Windows [Version 10.0.22631.4169]
(c) Microsoft Corporation. All rights reserved.

C:\DSCC>set path=C:\Program Files\Java\jdk1.8.0_241\bin

C:\DSCC>javac intfEqSolve.java

C:\DSCC>javac implEqSolve.java

C:\DSCC>start rmiregistry

C:\DSCC>|
```

```
C:\Windows\System32\cmd.e × + ▾ - □ ×
Microsoft Windows [Version 10.0.22631.4169]
(c) Microsoft Corporation. All rights reserved.

C:\DSCC>set path=C:\Program Files\Java\jdk1.8.0_241\bin

C:\DSCC>javac serverEqSolve.java

C:\DSCC>java serverEqSolve
|
```

```
C:\Windows\System32\cmd.e × + ▾ - □ ×
Microsoft Windows [Version 10.0.22631.4169]
(c) Microsoft Corporation. All rights reserved.

C:\DSCC>set path=C:\Program Files\Java\jdk1.8.0_241\bin

C:\DSCC>javac clientEqSolve.java

C:\DSCC>java clientEqSolve
Equations:-
1.(a-b)2
2.(a+b)2
3.(a-b)3
4.(a+b)3
Choose the equation:
1
Enter the value of a and b
19
18
the answer is1

C:\DSCC>|
```

Q6) RPC to implement server to print the string is palindrome

PalindromeStringServer.java:

```
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
public class PalindromeStringServer
{ private static final int PORT = 9876;
  public static void main(String[] args) {
    try (DatagramSocket socket = new DatagramSocket(PORT))
    { System.out.println("Palindrome String Server is running on port " +
PORT);
      byte[] receiveData = new byte[1024];
      while (true) {
        DatagramPacket receivePacket = new DatagramPacket(receiveData,
receiveData.length);
        socket.receive(receivePacket);
        String request = new String(receivePacket.getData(), 0,
receivePacket.getLength());
        InetAddress clientAddress = receivePacket.getAddress();
        int clientPort = receivePacket.getPort();
        String response = handleRequest(request);
        byte[] sendData = response.getBytes();
        DatagramPacket sendPacket = new DatagramPacket(sendData,
sendData.length, clientAddress, clientPort);
        socket.send(sendPacket);
      }
    } catch (Exception e)
    { e.printStackTrace();
    }
  }
  private static String handleRequest(String request)
  { return isPalindrome(request.trim()) ? "true" : "false";
  }
  private static boolean isPalindrome(String str) {
    String cleaned = str.replaceAll("\\s+", "").toLowerCase();
    String reversed = new StringBuilder(cleaned).reverse().toString();
    return cleaned.equals(reversed);
  }
}
```

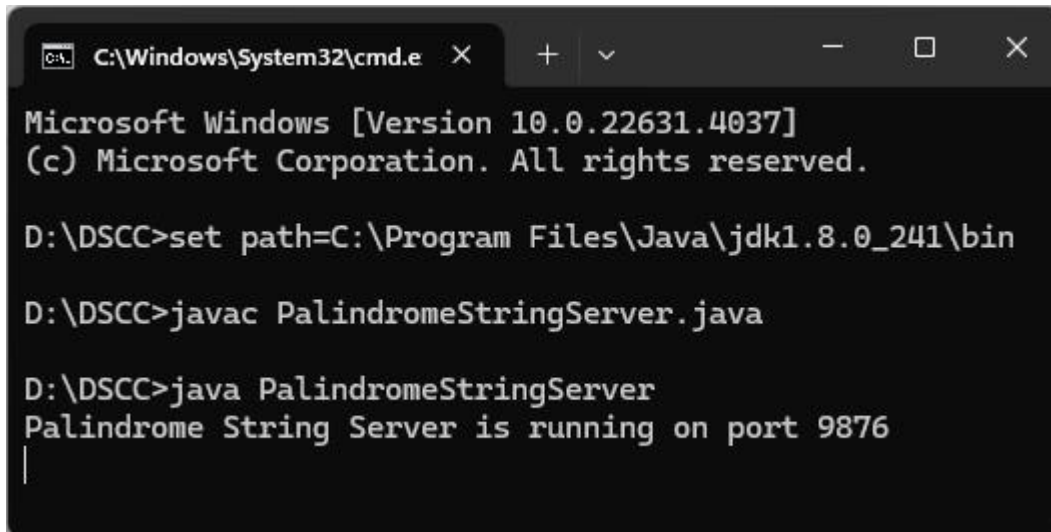
```
}  
}
```

PalindromeStringClient.java:

```
import java.net.DatagramPacket;  
import java.net.DatagramSocket;  
import java.net.InetAddress;  
import java.util.Scanner;  
public class PalindromeStringClient {  
    private static final String SERVER_ADDRESS = "localhost";  
    private static final int SERVER_PORT = 9876;  
    public static void main(String[] args) {  
        try (DatagramSocket socket = new DatagramSocket();  
            Scanner scanner = new Scanner(System.in))  
        {  
            InetAddress serverAddress =  
InetAddress.getByAddress(SERVER_ADDRESS);  
            while (true) {  
                System.out.println("Enter a string to check if it's a palindrome (or  
'exit' to quit):");  
                String input = scanner.nextLine();  
                if (input.equalsIgnoreCase("exit"))  
                    {break;  
                }  
                byte[] sendData = input.getBytes();  
                DatagramPacket sendPacket = new DatagramPacket(sendData,  
sendData.length, serverAddress, SERVER_PORT);  
                socket.send(sendPacket);  
                byte[] receiveData = new byte[1024];  
                DatagramPacket receivePacket = new DatagramPacket(receiveData,  
receiveData.length);  
                socket.receive(receivePacket);  
                String response = new String(receivePacket.getData(), 0,  
receivePacket.getLength());  
                System.out.println("Response from server: " +  
(response.equals("true") ? "Palindrome" : "Not a palindrome"));  
            }  
        } catch (Exception e) {
```

```
        e.printStackTrace();
    }
}
}
```

Output:

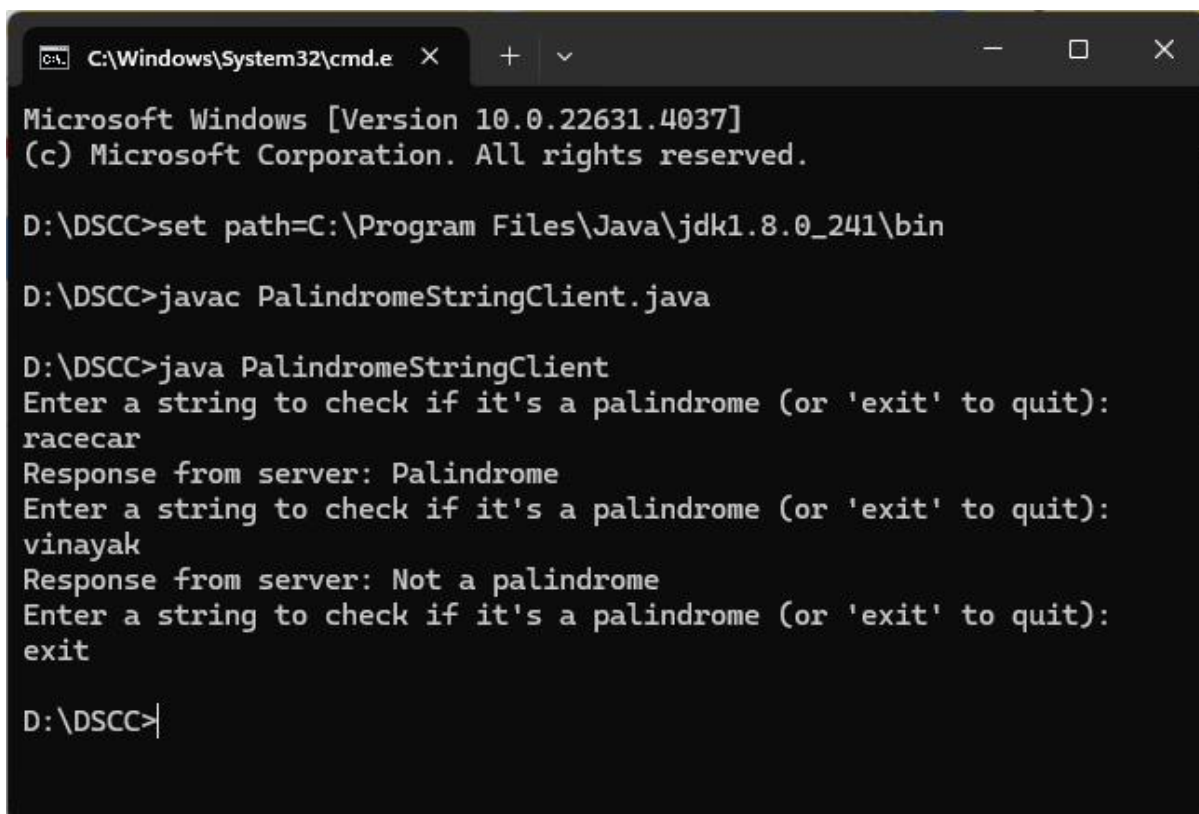


```
C:\Windows\System32\cmd.e X + v - □ X
Microsoft Windows [Version 10.0.22631.4037]
(c) Microsoft Corporation. All rights reserved.

D:\DSCC>set path=C:\Program Files\Java\jdk1.8.0_241\bin

D:\DSCC>javac PalindromeStringServer.java

D:\DSCC>java PalindromeStringServer
Palindrome String Server is running on port 9876
|
```



```
C:\Windows\System32\cmd.e X + v - □ X
Microsoft Windows [Version 10.0.22631.4037]
(c) Microsoft Corporation. All rights reserved.

D:\DSCC>set path=C:\Program Files\Java\jdk1.8.0_241\bin

D:\DSCC>javac PalindromeStringClient.java

D:\DSCC>java PalindromeStringClient
Enter a string to check if it's a palindrome (or 'exit' to quit):
racecar
Response from server: Palindrome
Enter a string to check if it's a palindrome (or 'exit' to quit):
vinayak
Response from server: Not a palindrome
Enter a string to check if it's a palindrome (or 'exit' to quit):
exit

D:\DSCC>|
```


Q7) RPC to implement server to print the if it is palindrome number

PalindromeServer.java:

```
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
public class PalindromeServer {
    private static final int PORT = 9876;
    public static void main(String[] args) {
        try (DatagramSocket socket = new DatagramSocket(PORT))
        { System.out.println("Palindrome Server is running on port " + PORT);
          byte[] receiveData = new byte[1024];
          while (true) {
              DatagramPacket receivePacket = new DatagramPacket(receiveData,
receiveData.length);
              socket.receive(receivePacket);
              String request = new String(receivePacket.getData(), 0,
receivePacket.getLength());
              InetAddress clientAddress = receivePacket.getAddress();
              int clientPort = receivePacket.getPort();
              String response = handleRequest(request);
              byte[] sendData = response.getBytes();
              DatagramPacket sendPacket = new DatagramPacket(sendData,
sendData.length, clientAddress, clientPort);
              socket.send(sendPacket);
          }
        } catch (Exception e)
        { e.printStackTrace();
        }
    }
    private static String handleRequest(String request)
    { try {
        int number = Integer.parseInt(request.trim());
        return isPalindrome(number) ? "true" : "false";
      } catch (NumberFormatException e)
      { return "Invalid number";
      }
    }
}
```

```

private static boolean isPalindrome(int number)
{
    String str = String.valueOf(number);
    String reversedStr = new StringBuilder(str).reverse().toString();
    return str.equals(reversedStr);
}
}

```

PalindromeClient.java:

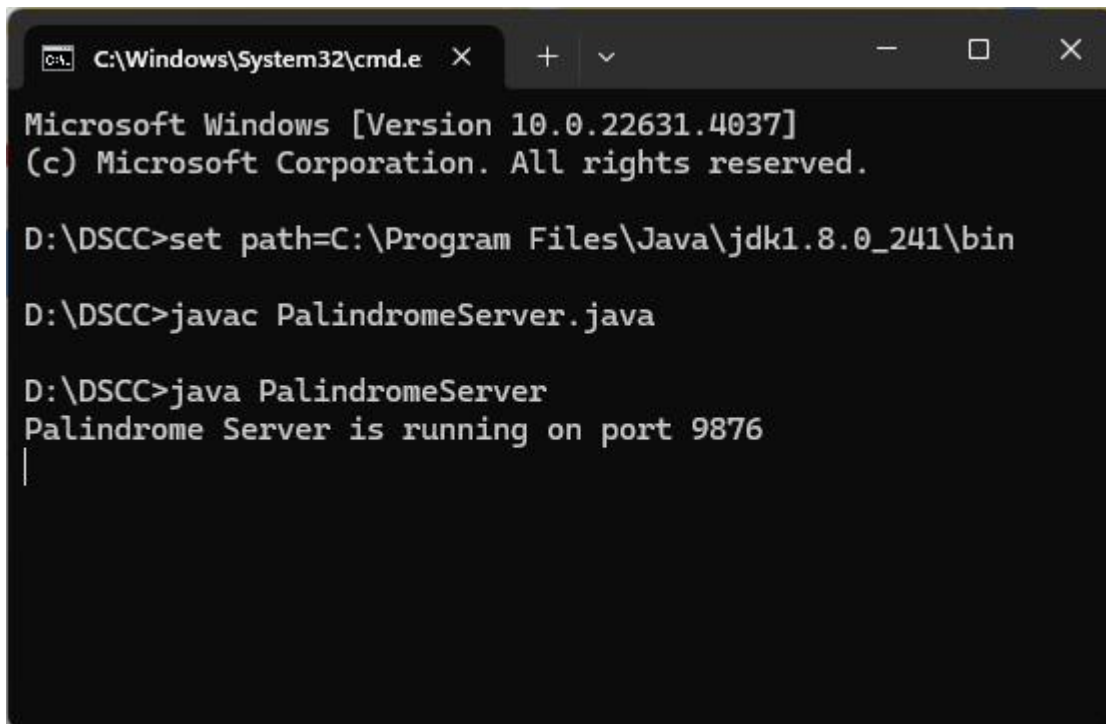
```

import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.util.Scanner;
public class PalindromeClient {
    private static final String SERVER_ADDRESS = "localhost";
    private static final int SERVER_PORT = 9876;
    public static void main(String[] args) {
        try (DatagramSocket socket = new DatagramSocket();
            Scanner scanner = new Scanner(System.in))
        {
            InetAddress serverAddress =
InetAddress.getByAddress(SERVER_ADDRESS);
            while (true) {
                System.out.println("Enter a number to check if it's a palindrome (or
'exit' to quit):");
                String input = scanner.nextLine();
                if (input.equalsIgnoreCase("exit"))
                {
                    break;
                }
                byte[] sendData = input.getBytes();
                DatagramPacket sendPacket = new DatagramPacket(sendData,
sendData.length, serverAddress, SERVER_PORT);
                socket.send(sendPacket);
                byte[] receiveData = new byte[1024];
                DatagramPacket receivePacket = new DatagramPacket(receiveData,
receiveData.length);
                socket.receive(receivePacket);
            }
        }
    }
}

```

```
        String response = new String(receivePacket.getData(), 0,
receivePacket.getLength());
        System.out.println("Response from server: " +
(response.equals("true") ? "Palindrome" : "Not a palindrome"));
    }
    } catch (Exception e)
    {e.printStackTrace();
    }
    }
}
```

Output:



```
C:\Windows\System32\cmd.e X + v - □ X
Microsoft Windows [Version 10.0.22631.4037]
(c) Microsoft Corporation. All rights reserved.

D:\DSCC>set path=C:\Program Files\Java\jdk1.8.0_241\bin

D:\DSCC>javac PalindromeServer.java

D:\DSCC>java PalindromeServer
Palindrome Server is running on port 9876
|
```

```
C:\Windows\System32\cmd.e X + v - □ X
Microsoft Windows [Version 10.0.22631.4037]
(c) Microsoft Corporation. All rights reserved.

D:\DSCC>set path=C:\Program Files\Java\jdk1.8.0_241\bin

D:\DSCC>javac PalindromeClient.java

D:\DSCC>java PalindromeClient
Error: Could not find or load main class PalindromeClient

D:\DSCC>javac PalindromeClient.java

D:\DSCC>java PalindromeClient
Enter a number to check if it's a palindrome (or 'exit' to quit):
1991
Response from server: Palindrome
Enter a number to check if it's a palindrome (or 'exit' to quit):
1919
Response from server: Not a palindrome
Enter a number to check if it's a palindrome (or 'exit' to quit):
exit

D:\DSCC>
```

Q8) Implement a Server containing factorial(), prime(), evenodd(),armstrong() etc using datagramsocket

MathServer.java:

```
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
public class MathServer {
    private static final int PORT = 9876;
    public static void main(String[] args) {
        try (DatagramSocket socket = new DatagramSocket(PORT))
        { System.out.println("Math Server is running on port " + PORT);
          byte[] receiveData = new byte[1024];
          while (true) {
              DatagramPacket receivePacket = new DatagramPacket(receiveData,
receiveData.length);
              socket.receive(receivePacket);
              String request = new String(receivePacket.getData(), 0,
receivePacket.getLength());
              InetAddress clientAddress = receivePacket.getAddress();
              int clientPort = receivePacket.getPort();
              String response = handleRequest(request);
              byte[] sendData = response.getBytes();
              DatagramPacket sendPacket = new DatagramPacket(sendData,
sendData.length, clientAddress, clientPort);
              socket.send(sendPacket);
          }
        } catch (Exception e)
        { e.printStackTrace();
        }
    }
    private static String handleRequest(String request)
    { String[] parts = request.split(" ");
      String command = parts[0];
      int number = Integer.parseInt(parts[1]);
      switch (command) {
          case "factorial":
              return String.valueOf(factorial(number));
```

```

        case "prime":
            return isPrime(number) ? "true" : "false";
        case "evenodd":
            return isEven(number) ? "even" : "odd";
        case "armstrong":
            return isArmstrong(number) ? "true" : "false";
        default:
            return "Unknown command";
    }
}

private static long factorial(int n)
{
    if (n == 0) return 1;
    return n * factorial(n - 1);
}

private static boolean isPrime(int n)
{
    if (n <= 1) return false;
    for (int i = 2; i <= Math.sqrt(n); i++)
        if (n % i == 0) return false;
    }
    return true;
}

private static boolean isEven(int n)
{
    return n % 2 == 0;
}

private static boolean isArmstrong(int n)
{
    int original = n;
    int result = 0;
    int digits = String.valueOf(n).length();
    while (n != 0) {
        int digit = n % 10;
        result += Math.pow(digit, digits);
        n /= 10;
    }
    return result == original;
}
}

```

MathClient.java:

```
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.util.Scanner;
public class MathClient {
    private static final String SERVER_ADDRESS = "localhost";
    private static final int SERVER_PORT = 9876;
    public static void main(String[] args) {
        try (DatagramSocket socket = new DatagramSocket();
            Scanner scanner = new Scanner(System.in))
        { InetAddress serverAddress =
InetAddress.getByName(SERVER_ADDRESS);
            while (true) {
                System.out.println("Enter command (factorial, prime, evenodd,
armstrong) and a number (or 'exit' to quit):");
                String input = scanner.nextLine();
                if (input.equalsIgnoreCase("exit"))
                    {break;
                }
                byte[] sendData = input.getBytes();
                DatagramPacket sendPacket = new DatagramPacket(sendData,
sendData.length, serverAddress, SERVER_PORT);
                socket.send(sendPacket);
                byte[] receiveData = new byte[1024];
                DatagramPacket receivePacket = new DatagramPacket(receiveData,
receiveData.length);
                socket.receive(receivePacket);
                String response = new String(receivePacket.getData(), 0,
receivePacket.getLength());
                System.out.println("Response from server: " + response);
            }
        } catch (Exception e)
        {e.printStackTrace();
        }
    }
}
```

Output:

```
C:\Windows\System32\cmd.e X + v - □ X
Microsoft Windows [Version 10.0.22631.4037]
(c) Microsoft Corporation. All rights reserved.

D:\DSCC>set path=C:\Program Files\Java\jdk1.8.0_241\bin

D:\DSCC>javac MathServer.java

D:\DSCC>java MathServer
Math Server is running on port 9876
|
```

```
C:\Windows\System32\cmd.e X + v - □ X
Microsoft Windows [Version 10.0.22631.4037]
(c) Microsoft Corporation. All rights reserved.

D:\DSCC>set path=C:\Program Files\Java\jdk1.8.0_241\bin

D:\DSCC>javac MathClient.java

D:\DSCC>java MathClient
Enter command (factorial, prime, evenodd, armstrong) and a number (or 'exit' to quit):
factorial 19
Response from server: 121645100408832000
Enter command (factorial, prime, evenodd, armstrong) and a number (or 'exit' to quit):
prime 19
Response from server: true
Enter command (factorial, prime, evenodd, armstrong) and a number (or 'exit' to quit):
evenodd 19
Response from server: odd
Enter command (factorial, prime, evenodd, armstrong) and a number (or 'exit' to quit):
armstrong 19
Response from server: false
Enter command (factorial, prime, evenodd, armstrong) and a number (or 'exit' to quit):
exit

D:\DSCC>|
```


Unit 3

Q1) The client should provide an equation to the server through an interface. The server will solve the expression given by the client.

ICalculator.java:

```
import java.rmi.*;
public interface ICalculator extends Remote
{
    double add(double x,double y) throws RemoteException;
    double sub(double x,double y) throws RemoteException;
    double mul(double x,double y) throws RemoteException;
    double div(double x,double y) throws RemoteException;
}
```

CalculatorImpl.java:

```
import java.rmi.*;
import java.rmi.server.*;
public class CalculatorImpl extends UnicastRemoteObject implements
ICalculator
{
    public CalculatorImpl() throws RemoteException
    {
    }
    public double add(double x,double y) throws RemoteException
    {
        return(x+y);
    }
    public double sub(double x,double y) throws RemoteException
    {
        return(x-y);
    }
    public double mul(double x,double y) throws RemoteException
    {return(x*y);
    }
    public double div(double x,double y) throws RemoteException
    {
        return(x/y);
    }
}
```

CalculatorServer.java:

```
import java.net.*;
import java.rmi.*;
public class CalculatorServer
{
    public static void main(String[] args)
    {
        try
        {
            CalculatorImpl ci=new CalculatorImpl();
            Naming.rebind("CalculatorServer",ci);
            System.out.println("Calculator Server is Ready");
        }
        catch(Exception e)
        {
            e.printStackTrace();
        }
    }
}
```

CalculatorClient.java:

```
import java.rmi.*;
import java.net.*;
import java.io.*;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
public class CalculatorClient extends JFrame implements ActionListener
{
    String strNum1="",strNum2="",strRes="",op="";
    double x,y,result;
    boolean flag,dotFlag,resFlag;
    ICalculator intf;
    GridBagConstraints gbc=new GridBagConstraints();
    JTextField txt1=new JTextField(20);
    JButton btn[]=new JButton[17];
    int i,j,k;
```

```

Container con;
public CalculatorClient()
{
con=this.getContentPane();
con.setLayout(new GridBagLayout());
gbc.weightx=1.0;
gbc.weighty=1.0;
btn[0]=new JButton("C");
btn[1]=new JButton("1");
btn[2]=new JButton("2");
btn[3]=new JButton("3");
btn[4]=new JButton("+");
btn[5]=new JButton("4");
btn[6]=new JButton("5");
btn[7]=new JButton("6");
btn[8]=new JButton("-");
btn[9]=new JButton("7");
btn[10]=new JButton("8");
btn[11]=new JButton("9");
btn[12]=new JButton("*");
btn[13]=new JButton("0");
btn[14]=new JButton(".");
btn[15]=new JButton("=");
btn[16]=new JButton("/");
gbc.gridx=0;
gbc.gridy=0;
gbc.gridwidth=4;
con.add(txt1,gbc);
gbc.gridwidth=1;
gbc.gridx=0;
gbc.gridy=1;
con.add(btn[0],gbc);
btn[0].addActionListener(this);i=1;
for(k=2;k<=5;k++)
{
for(j=0;j<=3;j++)

```

```

{
gbc.gridx=j;
gbc.gridy=k;
con.add(btn[i],gbc);
btn[i].addActionListener(this);
i++;
}
}
setSize(300,300);
setVisible(true);
}
public void actionPerformed(ActionEvent ae)
{
try
{
String url="rmi://127.0.0.1/CalculatorServer";
intf=(ICalculator)Naming.lookup(url);
}
catch(Exception e)
{
e.printStackTrace();
}
String cmd=ae.getActionCommand();
if(cmd.equals("C"))
{
txt1.setText("");
strNum1=strNum2=strRes="";
x=y=result=0;
flag=true;
dotFlag=false;
resFlag=false;
}
else if(cmd.equals("+") || cmd.equals("-") || cmd.equals("*")
||cmd.equals("/"))
{
if(flag)
{

```

```
strNum1=txt1.getText();
strNum2="";
x=Double.parseDouble(strNum1);
flag=false;
dotFlag=false;
resFlag=false;
}
txt1.setText("");
op=cmd;
}
else if(cmd.equals("="))
{
strNum2=txt1.getText();
y=Double.parseDouble(strNum2);
try
{
if(op.equals("+"))
{
result=intf.add(x,y);
}
if(op.equals("-"))
{
result=intf.sub(x,y);
}
if(op.equals("*"))
{
result=intf.mul(x,y);
}
if(op.equals("/"))
{
result=intf.div(x,y);
}
}
catch(Exception e)
{
e.printStackTrace();
}
```

```
txt1.setText(Double.toString(result));
dotFlag=true;
resFlag=true;
flag=true;
}
else if(cmd.equals("."))
{
if(!dotFlag)
{
txt1.setText(txt1.getText()+cmd);
dotFlag=true;
}
}
else
{
if(!resFlag)
{
txt1.setText(txt1.getText()+cmd);
}
}
}
public static void main(String[] args)
{
CalculatorClient cc=new CalculatorClient();
}
}
```

Output:

```
C:\Windows\System32\cmd.e  X + v - □ X
Microsoft Windows [Version 10.0.22631.4169]
(c) Microsoft Corporation. All rights reserved.

C:\DSCC>set path=C:\Program Files\Java\jdk1.8.0_241\bin

C:\DSCC>javac ICalculator.java

C:\DSCC>javac CalculatorImpl.java

C:\DSCC>start rmiregistry

C:\DSCC>|
```

```
C:\Windows\System32\cmd.e  X + v - □ X
Microsoft Windows [Version 10.0.22631.4169]
(c) Microsoft Corporation. All rights reserved.

C:\DSCC>set path=C:\Program Files\Java\jdk1.8.0_241\bin

C:\DSCC>javac CalculatorServer.java

C:\DSCC>java CalculatorServer
Calculator Server is Ready
|
```

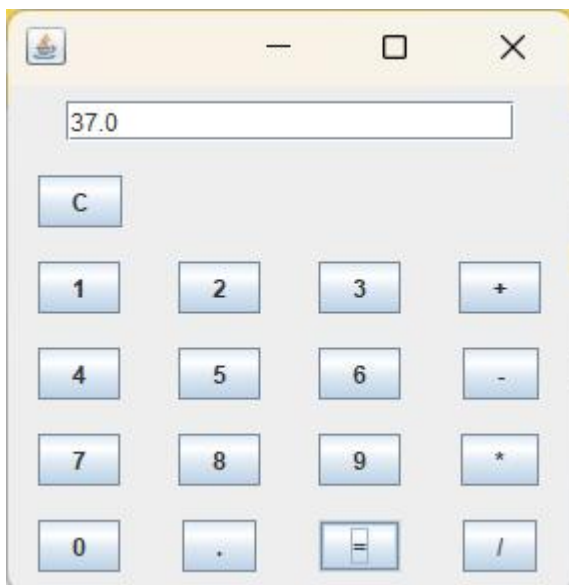
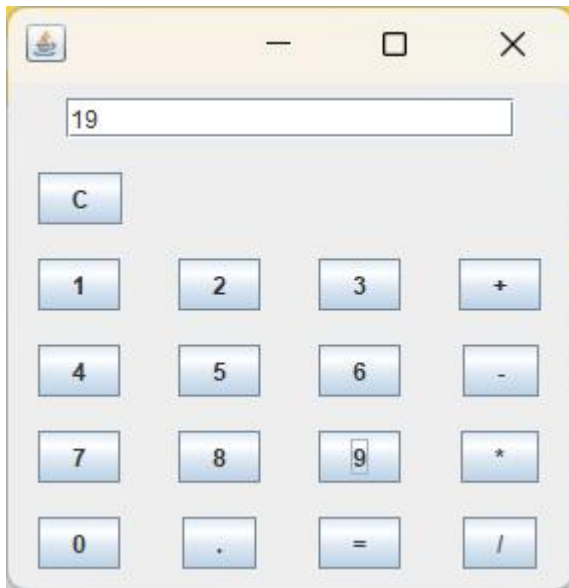
```
C:\Windows\System32\cmd.e  X + v - □ X
Microsoft Windows [Version 10.0.22631.4169]
(c) Microsoft Corporation. All rights reserved.

C:\DSCC>set path=C:\Program Files\Java\jdk1.8.0_241\bin

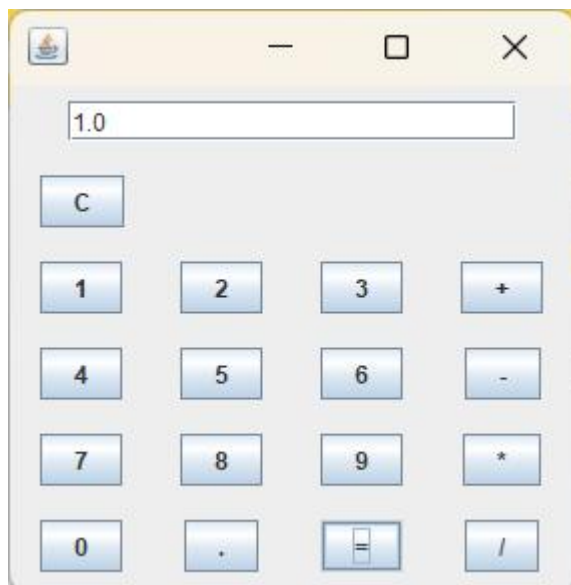
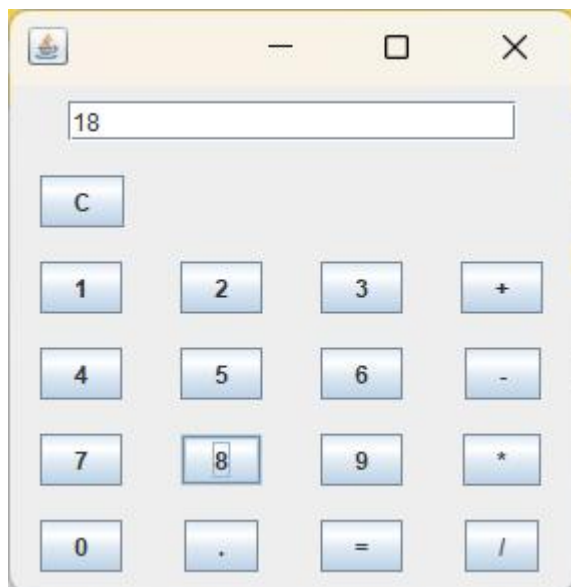
C:\DSCC>javac CalculatorClient.java

C:\DSCC>java CalculatorClient
```

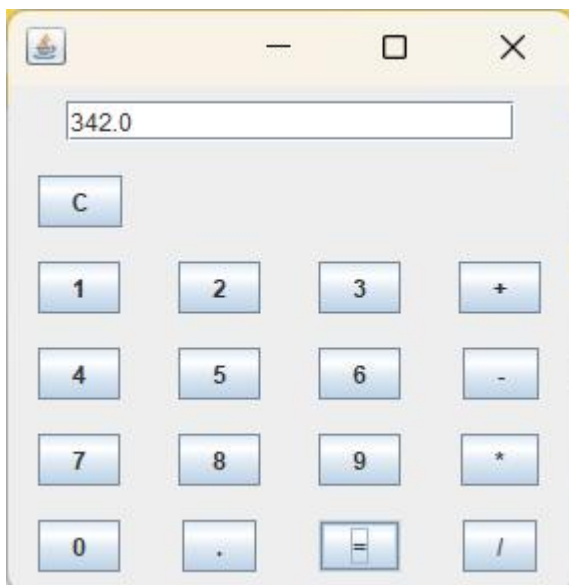

Add:



Subtract:



Multiply:



Divide:

