**Install java first :**

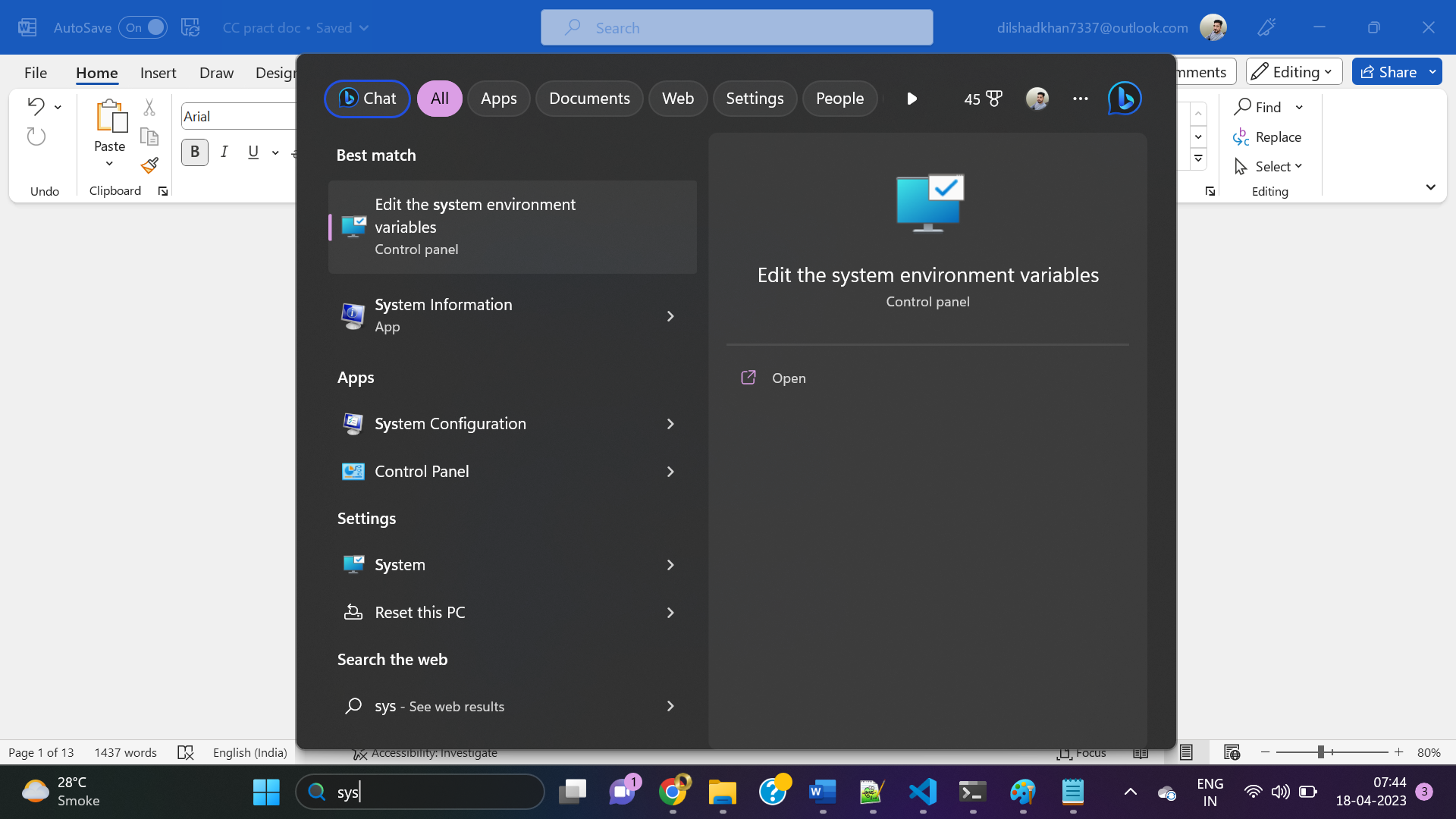
[**https://download.oracle.com/java/20/latest/jdk-20\_windows-x64\_bin.exe**](https://download.oracle.com/java/20/latest/jdk-20_windows-x64_bin.exe)

**install simply on clicking on next > next etc.**

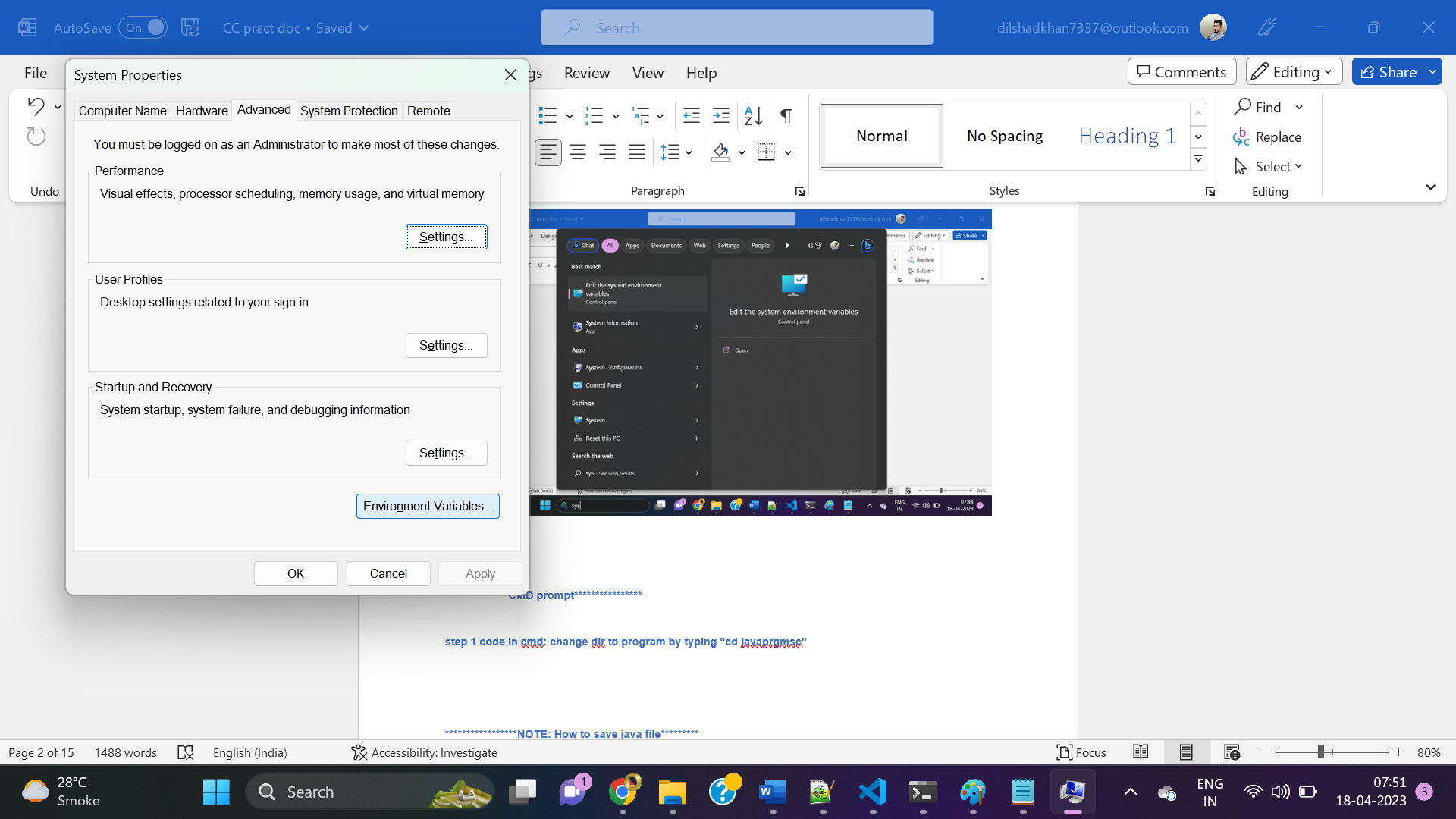
**now add the path for java programs in :**

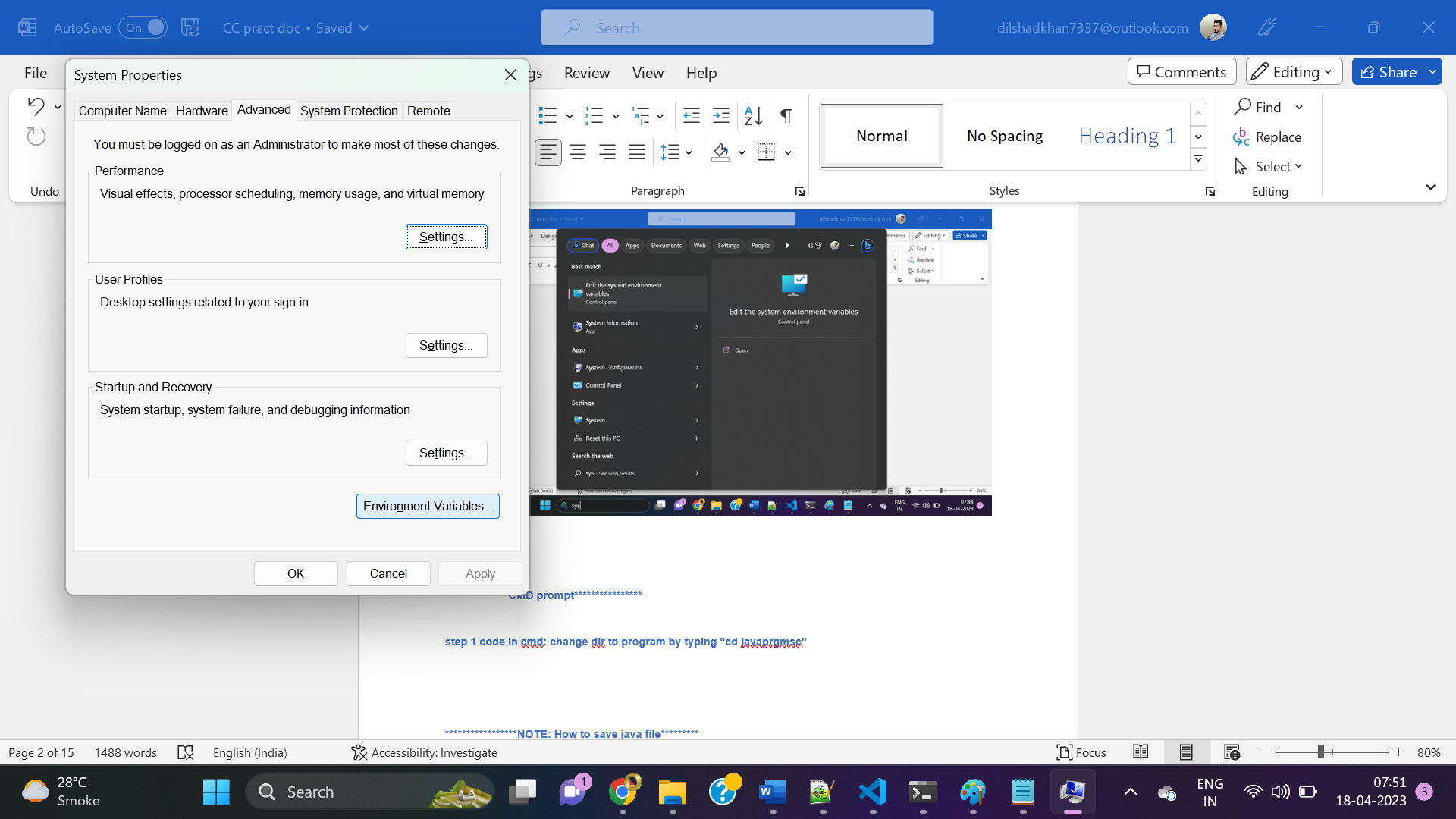
**Go to windows search > type edit the system variables as 1A screen> click on that and you will be able to see :**

**Environment variable as in 1B and follow screenshot**

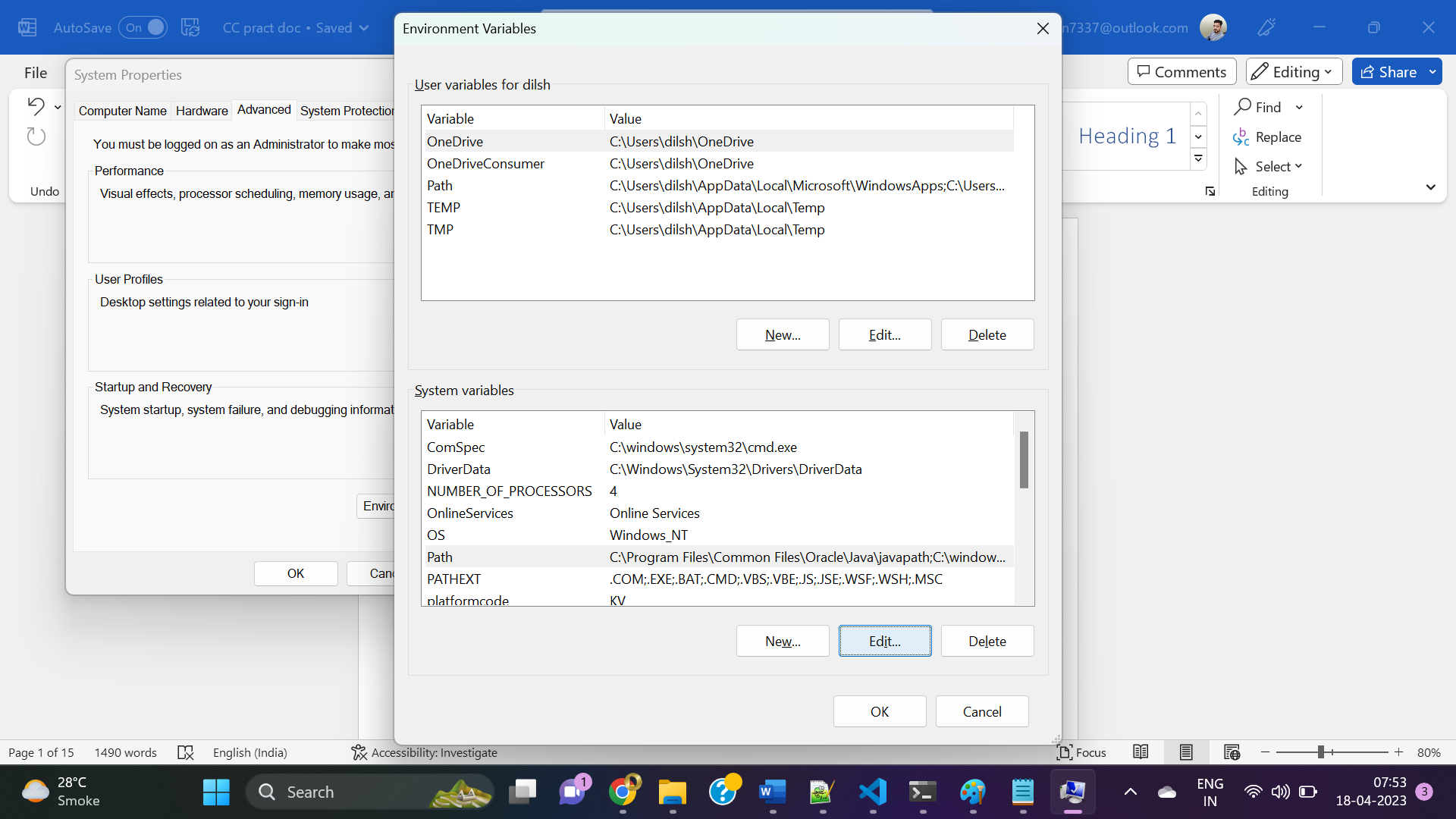


**1A**

 **1B**

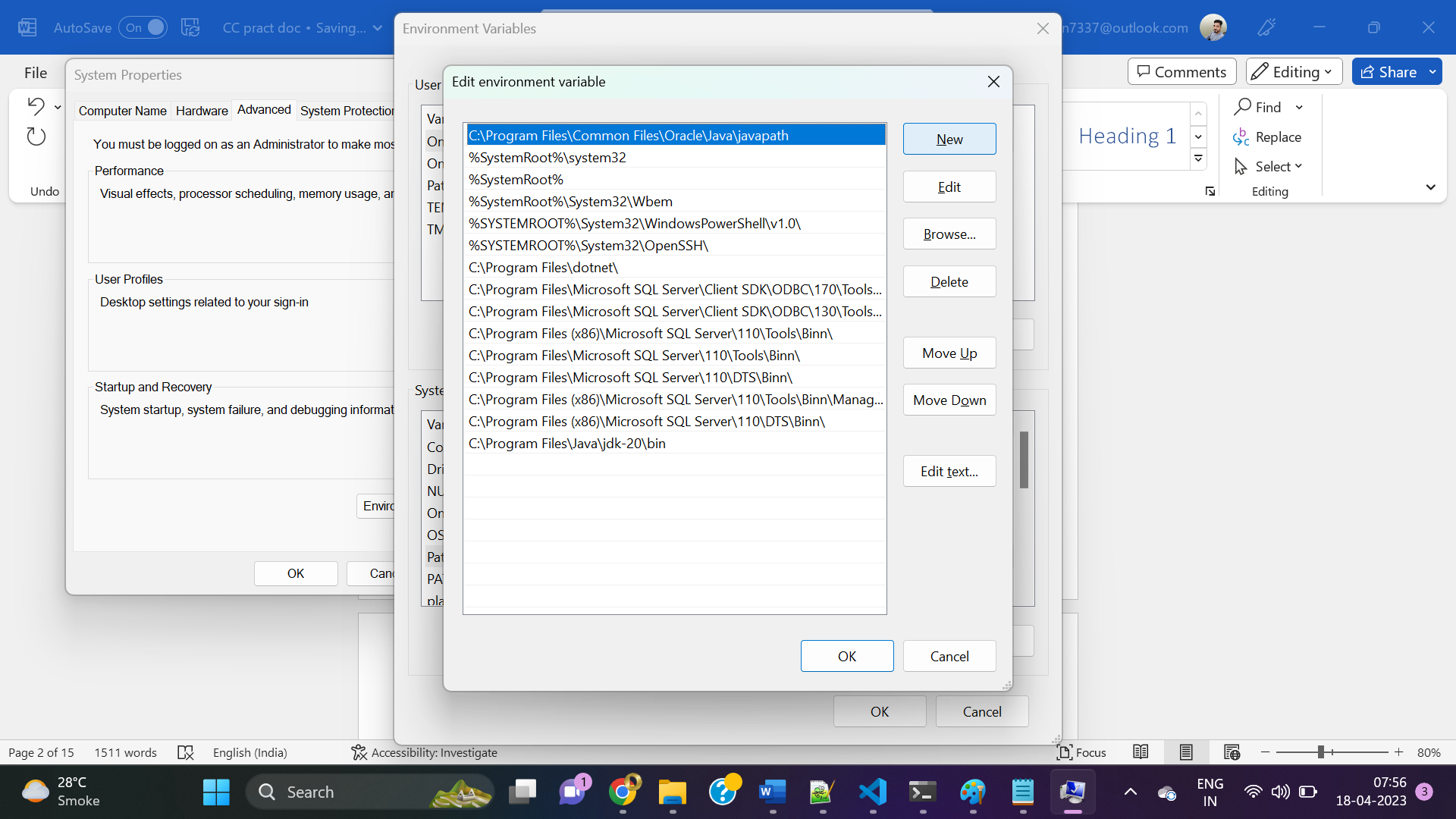


**1C**

 **1D**

**Now select path as screenshot and click on edit and you will see the below screen and click on**

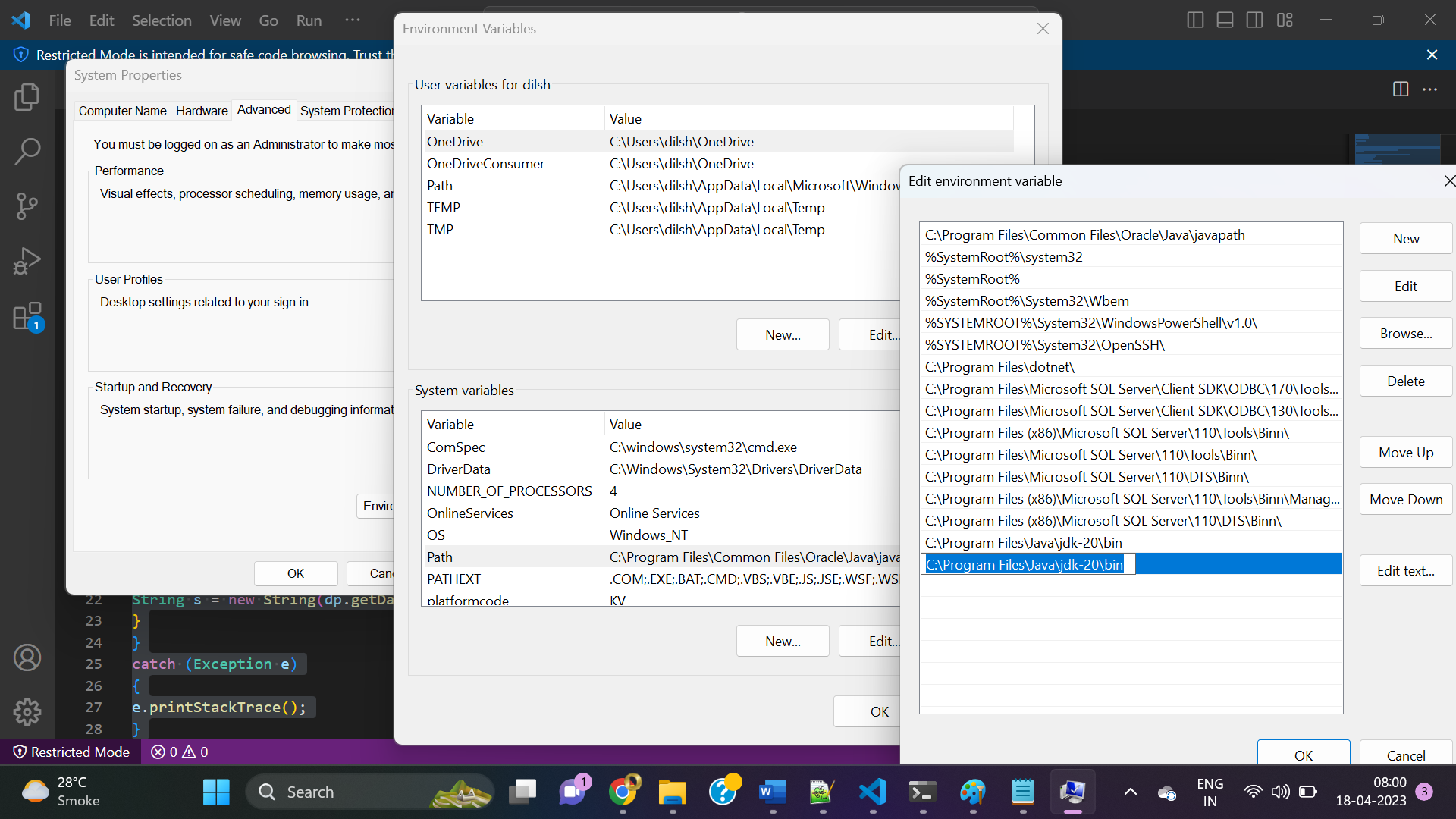
**New and you will se the below screen:**



**1E**

**Now go the path where java installed and copy the path of bin folder**

**As >>>> C:\Program Files\Java\jdk-20\bin**



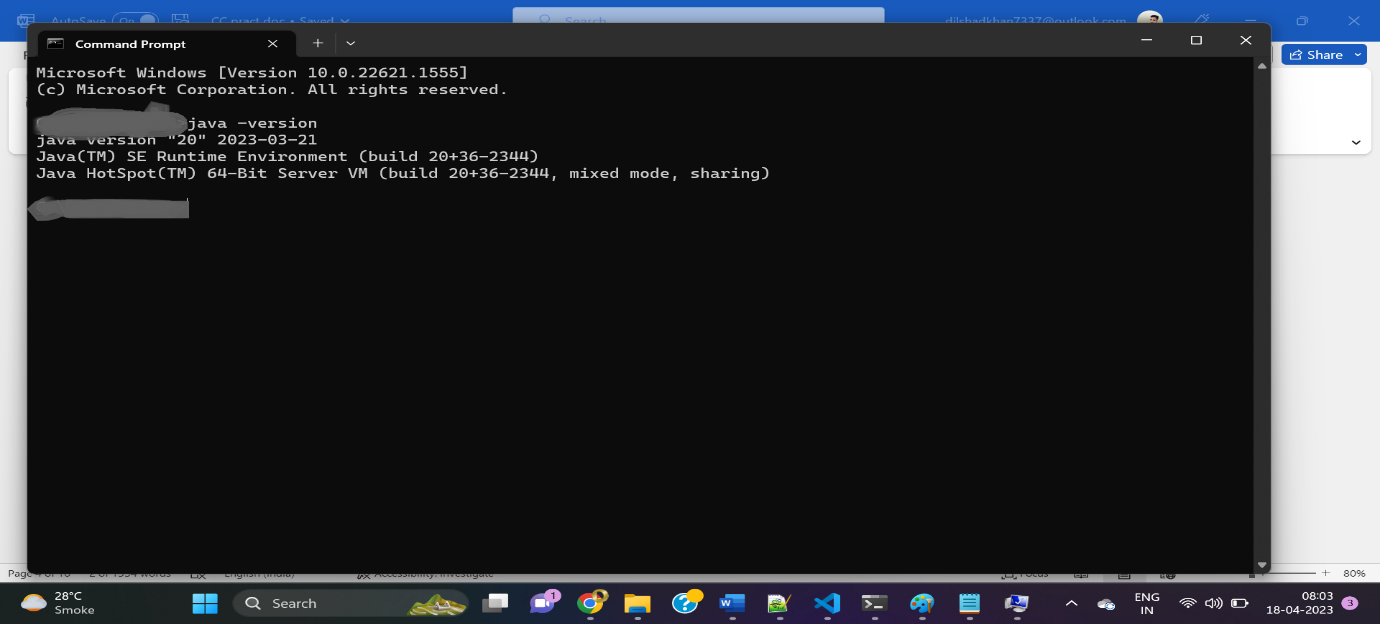
**And paste as above in the screen textbox and click ok on all opened popup screen.**

**Now go to CMD window**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*CMD prompt\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**step 1 check version by below syntax:**

**java -version**



**Always compile the java file after changing the directory in cmd windo** using **cd “path of the javafilefolder” where .java file saved**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*NOTE: How to save java file\*\*\*\*\*\*\*\*\***

**always save the java file with the class name.java inside the double code**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*CMD prompt\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**How to compile**

**after changing directory in cmd**

**type "**dir**" and** click enter **it will give the detail of all file name inside the javaprgmsc folder**

**javaprgmsc is my folder where I saved all java file**

**code for compiling :**

**"javac classname.java"**

**if no error then its ok**

**\*\*\*\*\*\*\*\*\*\*\*\*how to see output after succefull compilation\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**type:**

**java classname and click enter**

**1A) Write a program for implementing Client Server communication model using TCP.**

**/\* Save it like >>>> tcpServerPrime.java or always save with class name.java inside the double code\* /**

**/\* Program:\* /**

**import java.net.\*;**

**import java.io.\*;**

**class tcpServerPrime**

**{**

**public static void main(String args[])**

**{**

**try**

**{**

**ServerSocket ss = new ServerSocket(8001);**

**System.out.println("Server Started...............");**

**Socket s = ss.accept();**

**DataInputStream in = new DataInputStream(s.getInputStream()); int x= in.readInt();**

**DataOutputStream otc = new DataOutputStream(s.getOutputStream()); int y = x/2;**

**if(x ==1 || x ==2 || x ==3)**

**{**

**otc.writeUTF(x + "is Prime");**

**System.exit(0);**

**}**

**for(int i=2; i<=y; i++)**

**{**

**if(x%i != 0)**

**{**

**otc.writeUTF(x + " is Prime");**

**}**

**else**

**{**

**otc.writeUTF(x + " is not Prime");**

**}**

**}**

**}**

**catch(Exception e)**

**{**

**System.out.println(e.toString());**

**}**

**}**

**}**

**/\* Client side code:\* /  
  
import java.net.\*;**

**import java.io.\*;**

**class tcpClientPrime**

**{**

**public static void main(String args[])**

**{**

**try**

**{**

**Socket cs = new Socket("LocalHost",8001); BufferedReader infu = new BufferedReader(new**

**InputStreamReader(System.in));**

**System.out.println("Enter a number : ");**

**int a = Integer.parseInt(infu.readLine());**

**DataOutputStream out = new**

**DataOutputStream(cs.getOutputStream());**

**out.writeInt(a);**

**DataInputStream in = new DataInputStream(cs.getInputStream()); System.out.println(in.readUTF()); cs.close();**

**}**

**catch(Exception e)**

**{**

**System.out.println(e.toString());**

**}**

**}**

**}**

**Practical 1B: A client server TCP based chatting application.**

**ChatServer.java**

**Code :**

**import java.net.\*;**

**import java.io.\*;**

**class ChatServer**

**{**

**public static void main(String args[])**

**{**

**try**

**{**

**ServerSocket ss = new ServerSocket(8000);**

**System.out.println("Waiting for client to connect..");**

**Socket s = ss.accept();**

**BufferedReader br = new BufferedReader(new InputStreamReader(System.in));**

**DataOutputStream out = new DataOutputStream(s.getOutputStream());**

**DataInputStream in = new DataInputStream(s.getInputStream());**

**String receive, send;**

**while((receive = in.readLine()) != null)**

**{**

**if(receive.equals("STOP"))**

**break;**

**System.out.println("Client Says : "+receive);**

**System.out.print("Server Says : ");**

**send = br.readLine();**

**out.writeBytes(send+"\n");**

**}**

**br.close();**

**in.close();**

**out.close();**

**s.close();**

**}**

**catch(Exception e)**

**{**

**e.printStackTrace();**

**}**

**}**

**}**

**Save it with ChatClient.java**

**/\* ChatSide code \*/**

**import java.net.\*;**

**import java.io.\*;**

**class ChatClient**

**{**

**public static void main(String args[])**

**{**

**try**

**{**

**Socket s = new Socket("Localhost",8000);**

**BufferedReader br = new BufferedReader(new InputStreamReader(System.in));**

**DataOutputStream out = new DataOutputStream(s.getOutputStream());**

**DataInputStream in = new DataInputStream(s.getInputStream());**

**String msg;**

**System.out.println("To stop chatting with server type STOP");**

**System.out.print("Client Says: ");**

**while((msg = br.readLine()) != null)**

**{**

**out.writeBytes(msg+"\n");**

**if(msg.equals("STOP"))**

**break;**

**System.out.println("Server Says : "+in.readLine());**

**System.out.print("Client Says : ");**

**}**

**br.close();**

**in.close();**

**out.close();**

**s.close();**

**}**

**catch(Exception e)**

**{**

**e.printStackTrace();**

**}**

**}**

**}**

**2A) Write a program for implementing Client Server communication model using UDP.**

**/\*udpServerEO.java\*/**

**import java.io.\*;**

**import java.net.\*;**

**public class udpServerEO**

**{**

**public static void main(String args[])**

**{**

**try**

**{**

**DatagramSocket ds = new DatagramSocket(2000); byte b[] = new byte[1024];**

**DatagramPacket dp = new DatagramPacket(b,b.length); ds.receive(dp);**

**String str = new String(dp.getData(),0,dp.getLength()); System.out.println(str);**

**int a= Integer.parseInt(str); String s= new String();**

**if (a%2 == 0)**

**s = "Number is even";**

**else**

**s = "Number is odd";**

**byte b1[] = new byte[1024]; b1 = s.getBytes(); DatagramPacket dp1 = new**

**DatagramPacket(b1,b1.length,InetAddress.getLocalHost(),1000);**

**ds.send(dp1);**

**}**

**catch(Exception e)**

**{**

**e.printStackTrace();**

**}**

**}**

**}**

**/\* Client code:**

**Save as udpClientEO.java \*/**

**import java.io.\*;**

**import java.net.\*;**

**public class udpClientEO**

**{**

**public static void main(String args[])**

**{**

**try {**

**DatagramSocket ds = new DatagramSocket(1000); BufferedReader br = new BufferedReader(new**

**InputStreamReader(System.in));**

**System.out.println("Enter a number : "); String num = br.readLine();**

**byte b[] = new byte[1024]; b=num.getBytes(); DatagramPacket dp = new**

**DatagramPacket(b,b.length,InetAddress.getLocalHost(),2000);**

**ds.send(dp);**

**byte b1[] = new byte[1024];**

**DatagramPacket dp1 = new DatagramPacket(b1,b1.length); ds.receive(dp1);**

**String str = new String(dp1.getData(),0,dp1.getLength()); System.out.println(str);**

**}**

**catch(Exception e)**

**{**

**e.printStackTrace();**

**}**

**}**

**}**

**Practical 2B:** A client server based program using UDP to find the factorial of the entered number.

**/\* CODE : udpServerFact,java\*/**

**import java.io.\*;**

**import java.net.\*;**

**public class udpServerFact**

**{**

**public static void main(String args[])**

**{**

**try**

**{**

**DatagramSocket ds = new DatagramSocket(2000); byte b[] = new byte[1024];**

**DatagramPacket dp = new DatagramPacket(b,b.length); ds.receive(dp);**

**String str = new String(dp.getData(),0,dp.getLength()); System.out.println(str);**

**int a= Integer.parseInt(str); int f = 1, i;**

**String s= new String(); for(i=1;i<=a;i++)**

**{**

**f=f\*i;**

**}**

**s=Integer.toString(f);**

**String str1 = "The Factorial of " + str + " is : " + f; byte b1[] = new byte[1024]; b1 = str1.getBytes();**

**DatagramPacket dp1 = new DatagramPacket(b1,b1.length,InetAddress.getLocalHost(),1000);**

**ds.send(dp1);**

**}**

**catch(Exception e)**

**{**

**e.printStackTrace();**

**}**

**}**

**}**

**Client side code:**

**import java.io.\*;**

**import java.net.\*;**

**public class udpClientFact**

**{**

**public static void main(String args[])**

**{**

**try**

**{**

**DatagramSocket ds = new DatagramSocket(1000); BufferedReader br = new BufferedReader(new**

**InputStreamReader(System.in));**

**System.out.println("Enter a number : "); String num = br.readLine();**

**byte b[] = new byte[1024]; b=num.getBytes();**

**DatagramPacket dp = new DatagramPacket(b,b.length,InetAddress.getLocalHost(),2000);**

**ds.send(dp);**

**byte b1[] = new byte[1024];**

**DatagramPacket dp1 = new DatagramPacket(b1,b1.length); ds.receive(dp1);**

**String str = new String(dp1.getData(),0,dp1.getLength()); System.out.println(str);**

**}**

**catch(Exception e)**

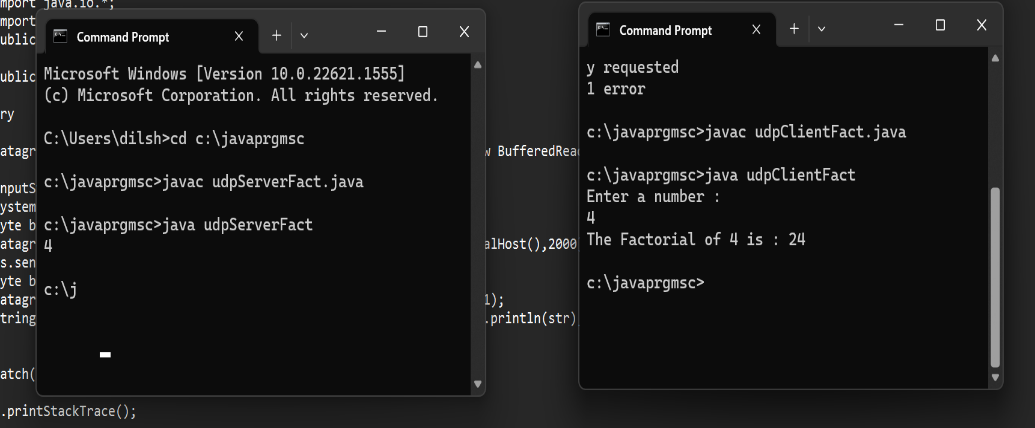
**{**

**e.printStackTrace();**

**}**

**}**

**}**

**Output :** 

**Practical 3C:** A program to implement simple calculator operations like addition, subtraction, multiplication and division.

**Code : RPCServer.java**

**import java.util.\*;**

**import java.net.\*;**

**class RPCServer**

**{**

**DatagramSocket ds;**

**DatagramPacket dp;**

**String str,methodName,result;**

**int val1,val2;**

**RPCServer()**

**{**

**try**

**{**

**ds=new DatagramSocket(1200);**

**byte b[]=new byte[4096];**

**while(true)**

**{**

**dp=new DatagramPacket(b,b.length);**

**ds.receive(dp);**

**str=new String(dp.getData(),0,dp.getLength());**

**if(str.equalsIgnoreCase("q"))**

**{**

**System.exit(1);**

**}**

**else**

**{**

**StringTokenizer st = new StringTokenizer(str," "); int i=0;**

**while(st.hasMoreTokens())**

**{**

**String token=st.nextToken();**

**methodName=token;**

**val1 = Integer.parseInt(st.nextToken());**

**val2 = Integer.parseInt(st.nextToken());**

**}**

**}**

**System.out.println(str);**

**InetAddress ia = InetAddress.getLocalHost();**

**if(methodName.equalsIgnoreCase("add"))**

**{**

**result= "" + add(val1,val2);**

**}**

**else if(methodName.equalsIgnoreCase("sub"))**

**{**

**result= "" + sub(val1,val2);**

**}**

**else if(methodName.equalsIgnoreCase("mul"))**

**{**

**result= "" + mul(val1,val2);**

**}**

**else if(methodName.equalsIgnoreCase("div"))**

**{**

**result= "" + div(val1,val2);**

**}**

**byte b1[]=result.getBytes();**

**DatagramSocket ds1 = new DatagramSocket(); DatagramPacket dp1 = new**

**DatagramPacket(b1,b1.length,InetAddress.getLocalHost(), 1300); System.out.println("result : "+result+"\n"); ds1.send(dp1);**

**}**

**}**

**catch (Exception e)**

**{**

**e.printStackTrace();**

**}**

**}**

**public int add(int val1, int val2)**

**{**

**return val1+val2;**

**}**

**public int sub(int val3, int val4)**

**{**

**return val3-val4;**

**}**

**public int mul(int val3, int val4)**

**{**

**return val3\*val4;**

**}**

**public int div(int val3, int val4)**

**{**

**return val3/val4;**

**}**

**public static void main(String[] args)**

**{**

**new RPCServer();**

**}**

**}**

**Code: RPCClient.java**

import java.io.\*;

import java.net.\*;

class RPCClient

{

RPCClient()

{

try

{

InetAddress ia = InetAddress.getLocalHost();

DatagramSocket ds = new DatagramSocket();

DatagramSocket ds1 = new DatagramSocket(1300);

System.out.println("\nRPC Client\n");

System.out.println("Enter method name and parameter like add 3 4\n");

while (true)

{

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

String str = br.readLine();

byte b[] = str.getBytes();

DatagramPacket dp = new

DatagramPacket(b,b.length,ia,1200);

ds.send(dp);

dp = new DatagramPacket(b,b.length);

ds1.receive(dp);

String s = new String(dp.getData(),0,dp.getLength()); System.out.println("\nResult = " + s + "\n");

}

}

catch (Exception e)

{

e.printStackTrace();

}

}

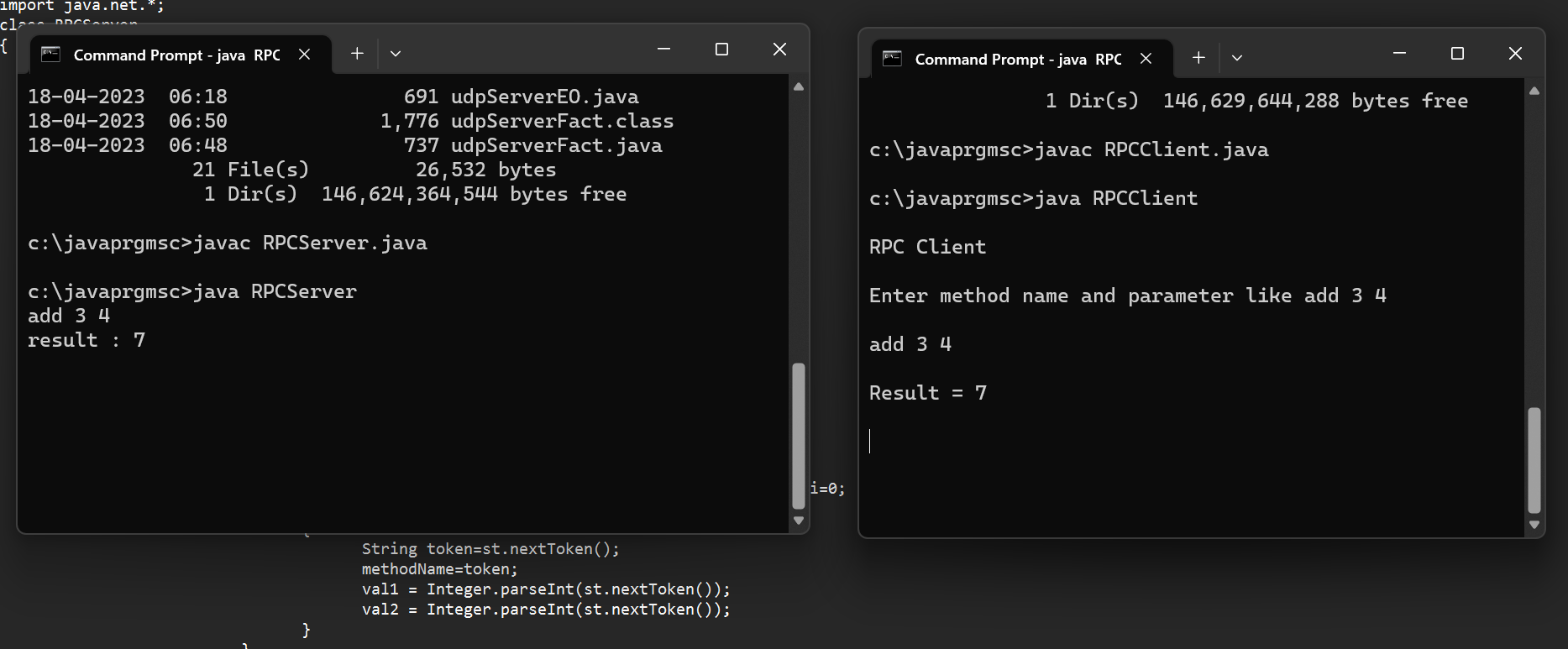
public static void main(String[] args)

{

new RPCClient();

}

}

**Output**

**Practical : A program that finds the square, square root, cube and cube root of the entered number.**

Code: **RPCNumServer.java**

**import java.util.\*;**

**import java.net.\*;**

**import java.io.\*;**

**class RPCNumServer**

**{**

**DatagramSocket ds;**

**DatagramPacket dp;**

**String str,methodName,result;**

**int val;**

**RPCNumServer()**

**{**

**try**

**{**

**ds=new DatagramSocket(1200);**

**byte b[]=new byte[4096];**

**while(true)**

**{**

**dp=new DatagramPacket(b,b.length);**

**ds.receive(dp);**

**str=new String(dp.getData(),0,dp.getLength());**

**if(str.equalsIgnoreCase("q")) {**

**System.exit(1);**

**}**

**else**

**{**

**StringTokenizer st = new StringTokenizer(str," ");**

**int i=0;**

**while(st.hasMoreTokens())**

**{**

**String token=st.nextToken();**

**methodName=token;**

**val = Integer.parseInt(st.nextToken());**

**}**

**}**

**System.out.println(str);**

**InetAddress ia = InetAddress.getLocalHost();**

**if(methodName.equalsIgnoreCase("square"))**

**{**

**result= "" + square(val);**

**}**

**else if(methodName.equalsIgnoreCase("squareroot"))**

**{**

**result= "" + squareroot(val);**

**}**

**else if(methodName.equalsIgnoreCase("cube"))**

**{**

**result= "" + cube(val);**

**}**

**else if(methodName.equalsIgnoreCase("cuberoot"))**

**{**

**result= "" + cuberoot(val);**

**}**

**byte b1[]=result.getBytes();**

**DatagramSocket ds1 = new DatagramSocket();**

**DatagramPacket dp1 = new**

**DatagramPacket(b1,b1.length,InetAddress.getLocalHost(), 1300); System.out.println("result : "+result+"\n"); ds1.send(dp1);**

**}**

**}**

**catch (Exception e)**

**{**

**e.printStackTrace();**

**}**

**}**

**public double square(int a) throws Exception**

**{**

**double ans;**

**ans = a\*a;**

**return ans;**

**}**

**public double squareroot(int a) throws Exception**

**{**

**double ans;**

**ans = Math.sqrt(a);**

**return ans;**

**}**

**public double cube(int a) throws Exception**

**{**

**double ans;**

**ans = a\*a\*a;**

**return ans;**

**}**

**public double cuberoot(int a) throws Exception**

**{**

**double ans;**

**ans = Math.cbrt(a);**

**return ans;**

**}**

**public static void main(String[] args)**

**{**

**new RPCNumServer();**

**}**

**}**

**CODE : RPCNumClient.java**

**import java.io.\*;**

**import java.net.\*;**

**class RPCNumClient**

**{**

**RPCNumClient()**

**{**

**try**

**{**

**InetAddress ia = InetAddress.getLocalHost(); DatagramSocket ds = new DatagramSocket(); DatagramSocket ds1 = new DatagramSocket(1300); System.out.println("\nRPC Client\n");**

**System.out.println("1. Square of the number - square\n2. Square root of the number - squareroot\n3. Cube of the number - cube\n4. Cube root of the number - cuberoot");**

**System.out.println("Enter method name and the number\n");**

**while (true)**

**{**

**BufferedReader br = new BufferedReader(new InputStreamReader(System.in));**

**String str = br.readLine();**

**byte b[] = str.getBytes();**

**DatagramPacket dp = new**

**DatagramPacket(b,b.length,ia,1200);**

**ds.send(dp);**

**dp = new DatagramPacket(b,b.length);**

**ds1.receive(dp);**

**String s = new String(dp.getData(),0,dp.getLength()); System.out.println("\nResult = " + s + "\n");**

**}**

**}**

**catch (Exception e)**

**{**

**e.printStackTrace();**

**}**

**}**

**public static void main(String[] args)**

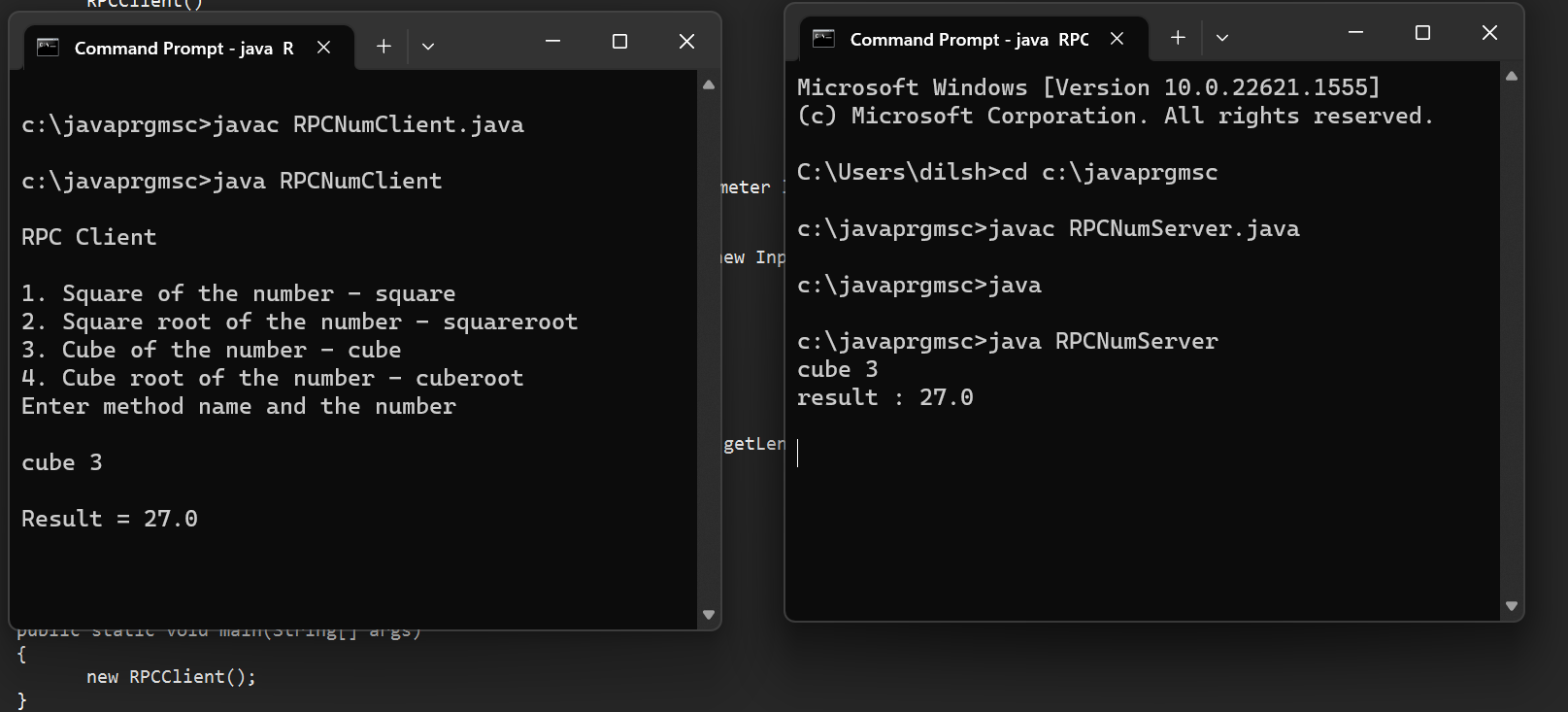
**{**

**new RPCNumClient();**

**}**

**}**

**OuTPUT:**



**Practical No: 03**

**Quest:** **A multicast Socket example.**

**Code: BroadcastServer.java**

**import java.net.\*;**

**import java.io.\*;**

**import java.util.\*;**

**public class BroadcastServer**

**{**

**public static final int PORT = 1234;**

**public static void main(String args[])throws**

**Exception**

**{**

**MulticastSocket socket;**

**DatagramPacket packet;**

**InetAddress address;**

**// set the multicast address to your local subnet**

**address = InetAddress.getByName("239.1.2.3");**

**socket = new MulticastSocket();**

**// join a Multicast group and send the group messages**

**socket.joinGroup(address);**

**byte[] data = null;**

**for(;;)**

**{**

**Thread.sleep(10000);**

**System.out.println("Sending ");**

**String str = ("This is Pushpa Calling....");**

**data = str.getBytes();**

**packet = new DatagramPacket(data, str.length(),address,PORT);**

**// Sends the packet**

**socket.send(packet);**

**}**

**}**

**}**

**Code: BroadcastClient.java**

**import java.net.\*;**

**import java.io.\*;**

**public class BroadcastClient**

**{**

**public static final int PORT = 1234;**

**public static void main(String args[])throws Exception**

**{**

**MulticastSocket socket;**

**DatagramPacket packet;**

**InetAddress address;**

**// set the mulitcast address to your local subnet**

**address = InetAddress.getByName("239.1.2.3");**

**socket = new MulticastSocket(PORT);**

**//join a Multicast group and wait for a message**

**socket.joinGroup(address);**

**byte[] data = new byte[100];**

**packet = new DatagramPacket(data,data.length);**

**for(;;)**

**{**

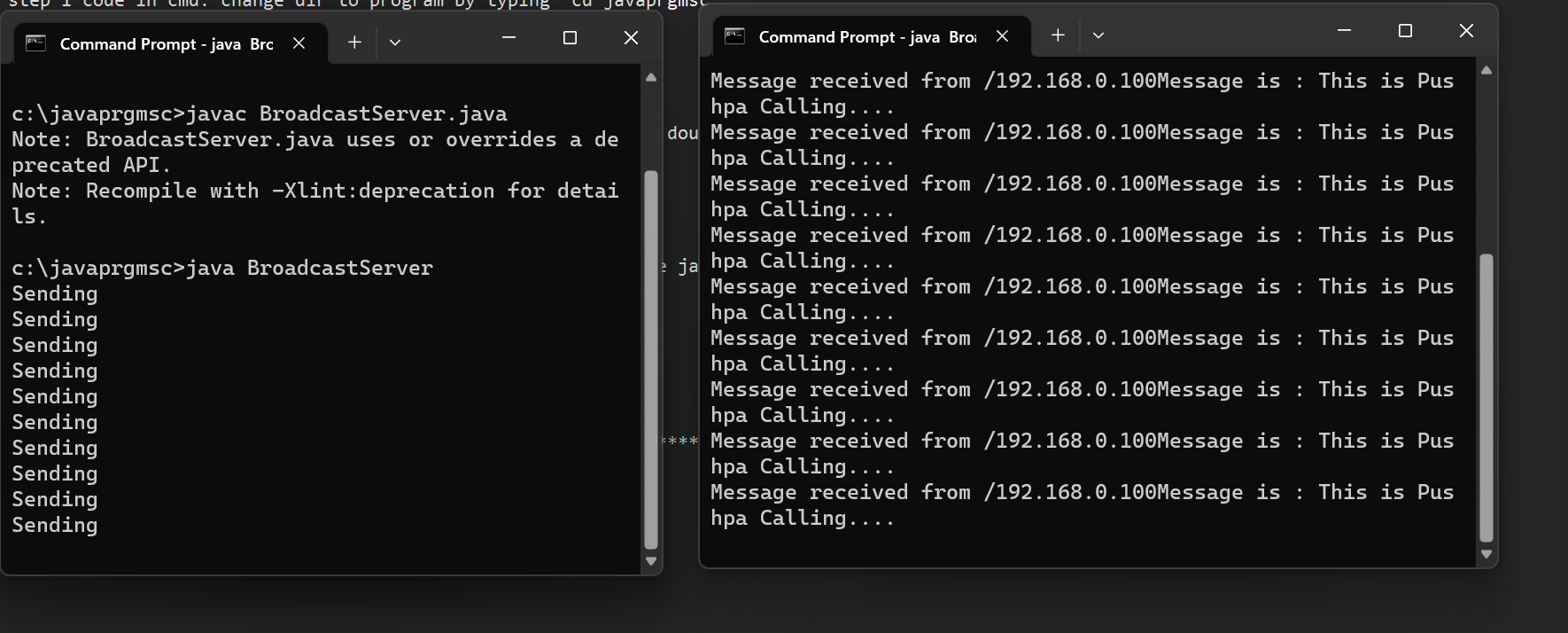
**socket.receive(packet);**

**String str = new String(packet.getData());**

**System.out.println("Message received from "+ packet.getAddress() + "Message is : "+str);**

**}**

**}**

**}Output:**

**Practical No: 04**

Question: A RMI based application program to display current date and time.