**Experiment no – 01**

**Aim: Demonstrate Synchronization Mechanism in Process Management using threads.**

**Theory: -**

**Program: -**

**Server.java**

import java.io.\*;

import java.net.\*;

class Server {

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

        ServerSocket ss = new ServerSocket(2222);

        Socket s = ss.accept();

        InputStreamReader isr = new InputStreamReader(System.in);

        BufferedReader keyboard = new BufferedReader(isr);

        BufferedReader br1 = new BufferedReader(new

InputStreamReader(s.getInputStream()));

        PrintWriter pw = new PrintWriter((s.getOutputStream()));

        while (true) {

            System.out.println("Enter Send/Quit/Receive");

            String k = keyboard.readLine();

            switch (k.charAt(0)) {

                case 'S':

                case 's':

                    String sendmsg = keyboard.readLine();

                    pw.println(sendmsg);

                    pw.flush();

                    break;

                case 'R':

                case 'r':

                    String recvmsg = br1.readLine();

                    System.out.println(recvmsg);

                    break;

                case 'Q':

                case 'q':

                    System.exit(0);

            }

        }

    }

}

**Client.java**

import java.io.\*;

import java.net.\*;

class Client {

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

        InputStreamReader isr = new InputStreamReader(System.in);

        BufferedReader keyboard = new BufferedReader(isr);

        Socket s = new Socket("localhost", 2222);

        BufferedReader br1 = new BufferedReader(new InputStreamReader(s.getInputStream()));

        PrintWriter pw = new PrintWriter((s.getOutputStream()));

        while (true) {

            System.out.println("Enter Send/Quit/Receive");

            String k = keyboard.readLine();

            switch (k.charAt(0)) {

                case 'S':

                case 's':

                    String sendmsg = keyboard.readLine();

                    pw.println(sendmsg);

                    pw.flush();

                    break;

                case 'R':

                case 'r':

                    String recvmsg = br1.readLine();

                    System.out.println(recvmsg);

                    break;

                case 'Q':

                case 'q':

                    System.exit(0);

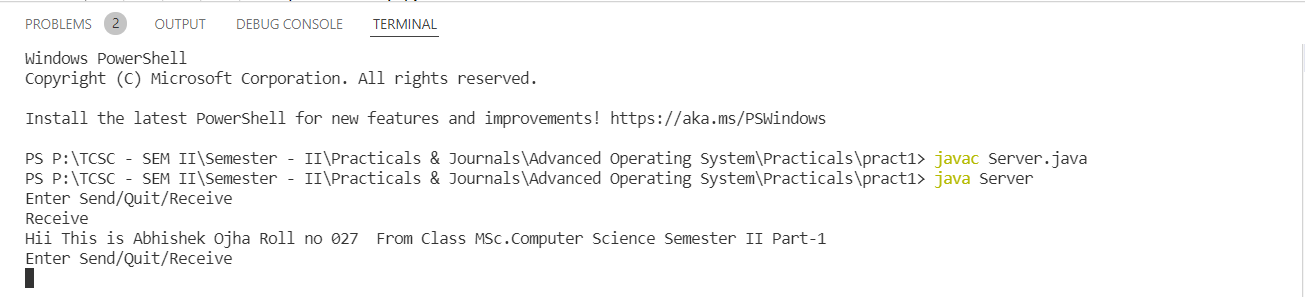
            }

        }

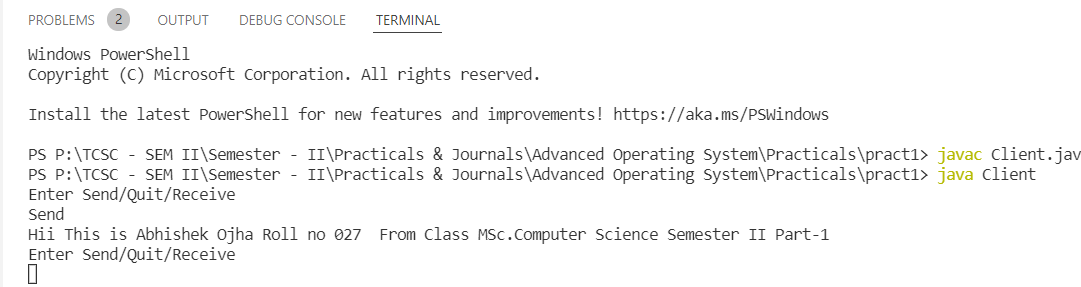
    }

}

**Output:**

**Server**

**Client**



**Myprogram.java**

class  ABC

{

    synchronized void print()

    {   int a=0;

       System.out.println("hi");

       System.out.println("I");

       System.out.println("am");

       ++a;

       try{

       Thread.sleep(500);

       }

       catch(Exception e)

        {

        }

       System.out.println("Abhishek Ojha Roll No 027 From MS.CS Learning"+a);

       System.out.println("Java");

       }

}

class myprog extends Thread

{  ABC ob1;

myprog(ABC ob)

    {

    ob1=ob;

    }

public void run()

    {

    ob1.print();

    }

}

class mainprog

{

  public static void main(String args[])

    {

    ABC ob = new ABC();

    myprog t1=new myprog(ob);

        myprog t2=new myprog(ob);

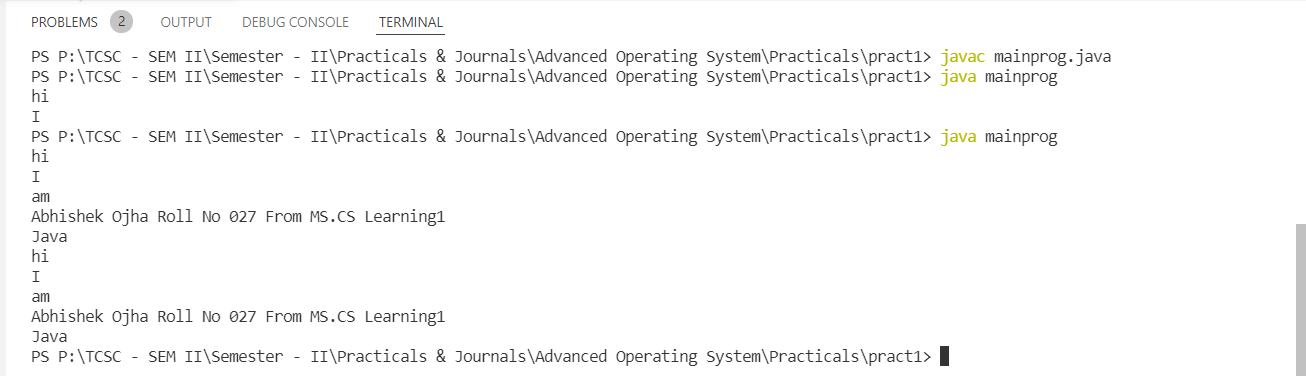
        t1.start();

        t2.start();

    }

}

**Output:**

****