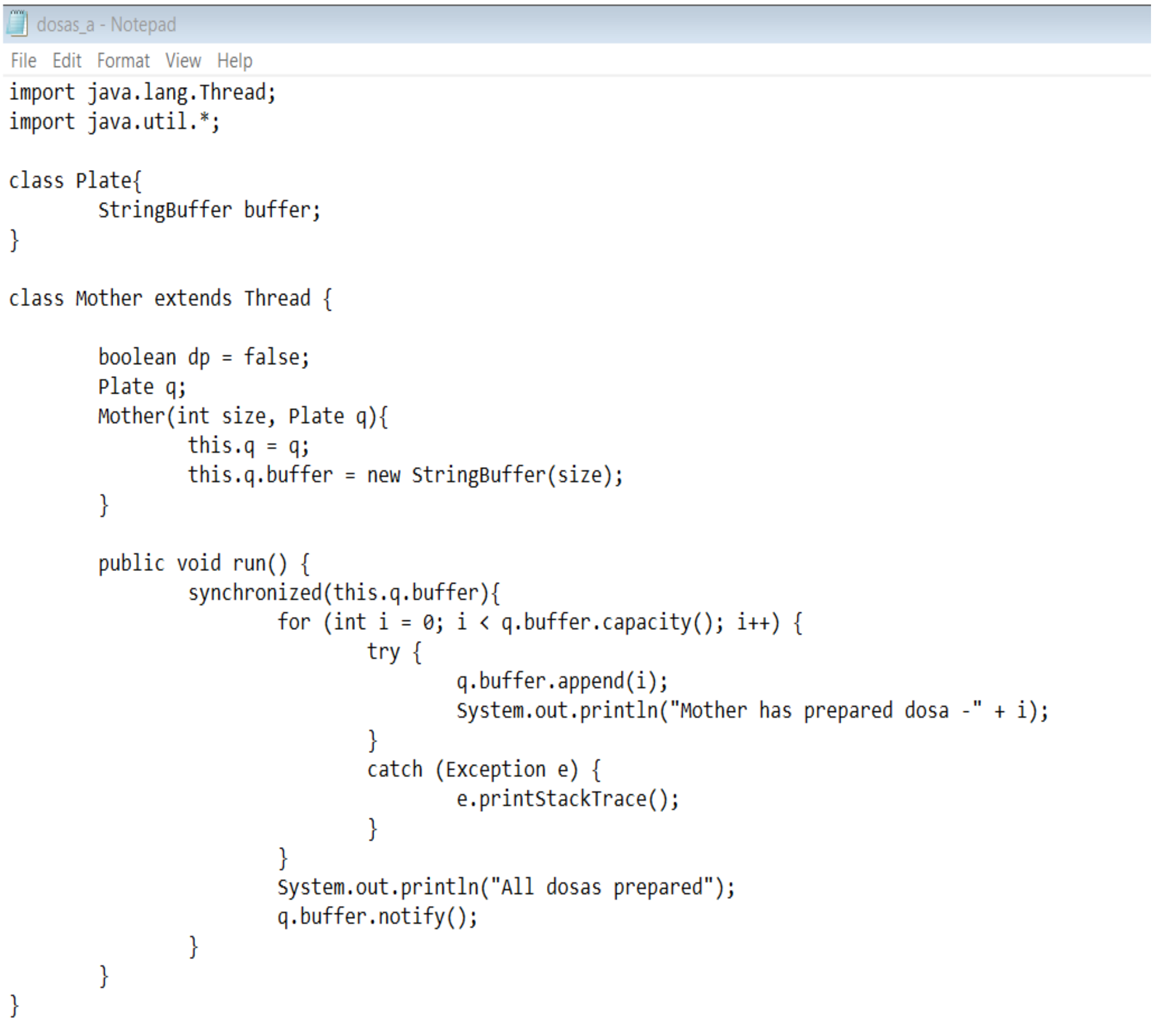
WEEK 9 EXERCISE-8 JAVA PROGRAMMING

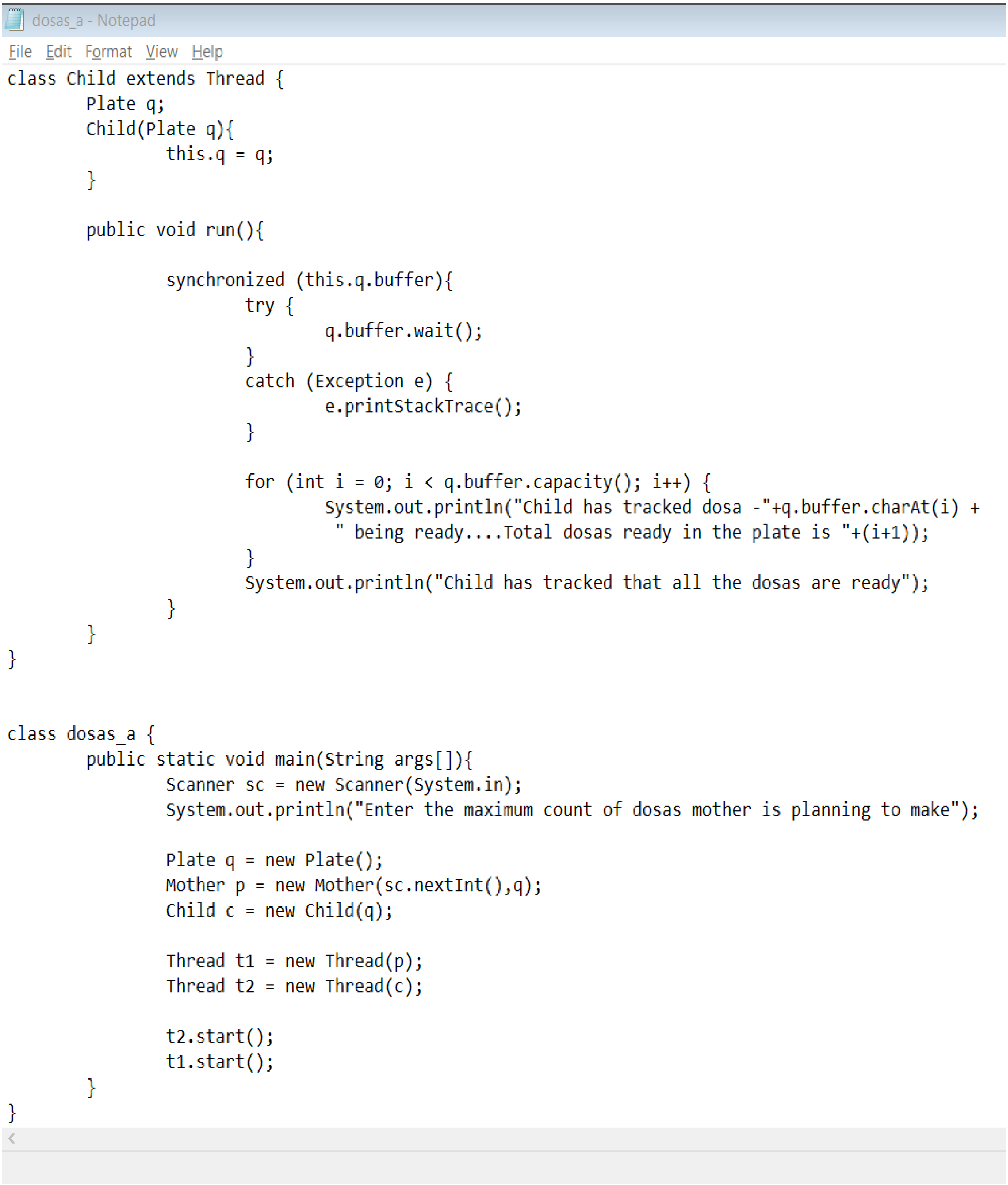
MULTITHREADING AND NETWORKING IN JAVA

S.SRIHARI (2018103601)

Q1. MULTITHREADING IN JAVA

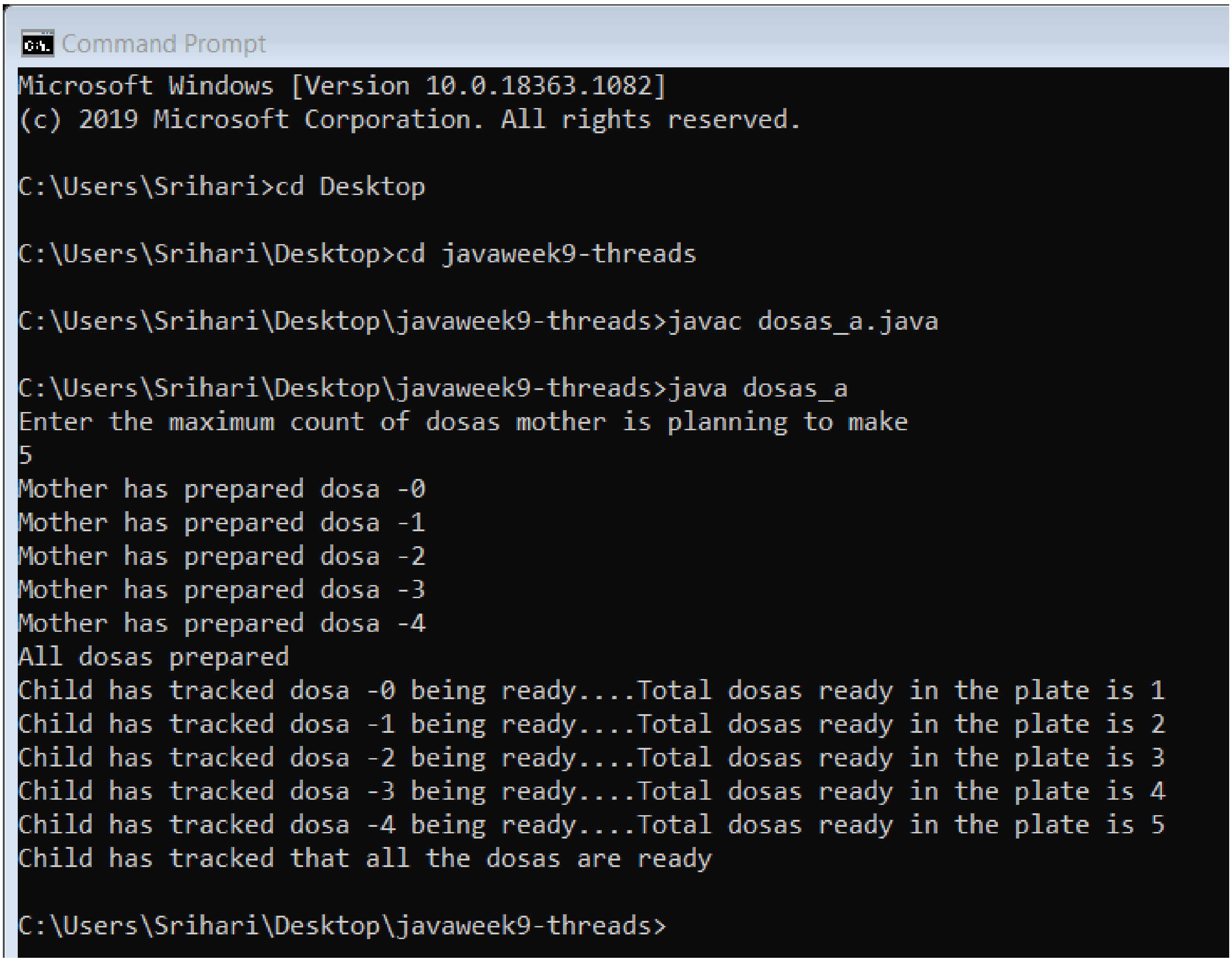


```
import java.lang.Thread;
import java.util.*;

class Plate{
        StringBuffer buffer;
}

class Mother extends Thread {

        boolean dp = false;
        Plate q;
        Mother(int size, Plate q){
                this.q = q;
                this.q.buffer = new StringBuffer(size);
        }

        public void run() {
                synchronized(this.q.buffer){
                        for (int i = 0; i < q.buffer.capacity(); i++) {
                                try {
                                        q.buffer.append(i);
                                        System.out.println("Mother has prepared dosa -" + i);
                                }
                                catch (Exception e) {
                                        e.printStackTrace();
                                }
                        }
                        System.out.println("All dosas prepared");
                        q.buffer.notify();
                }
        }
}
```



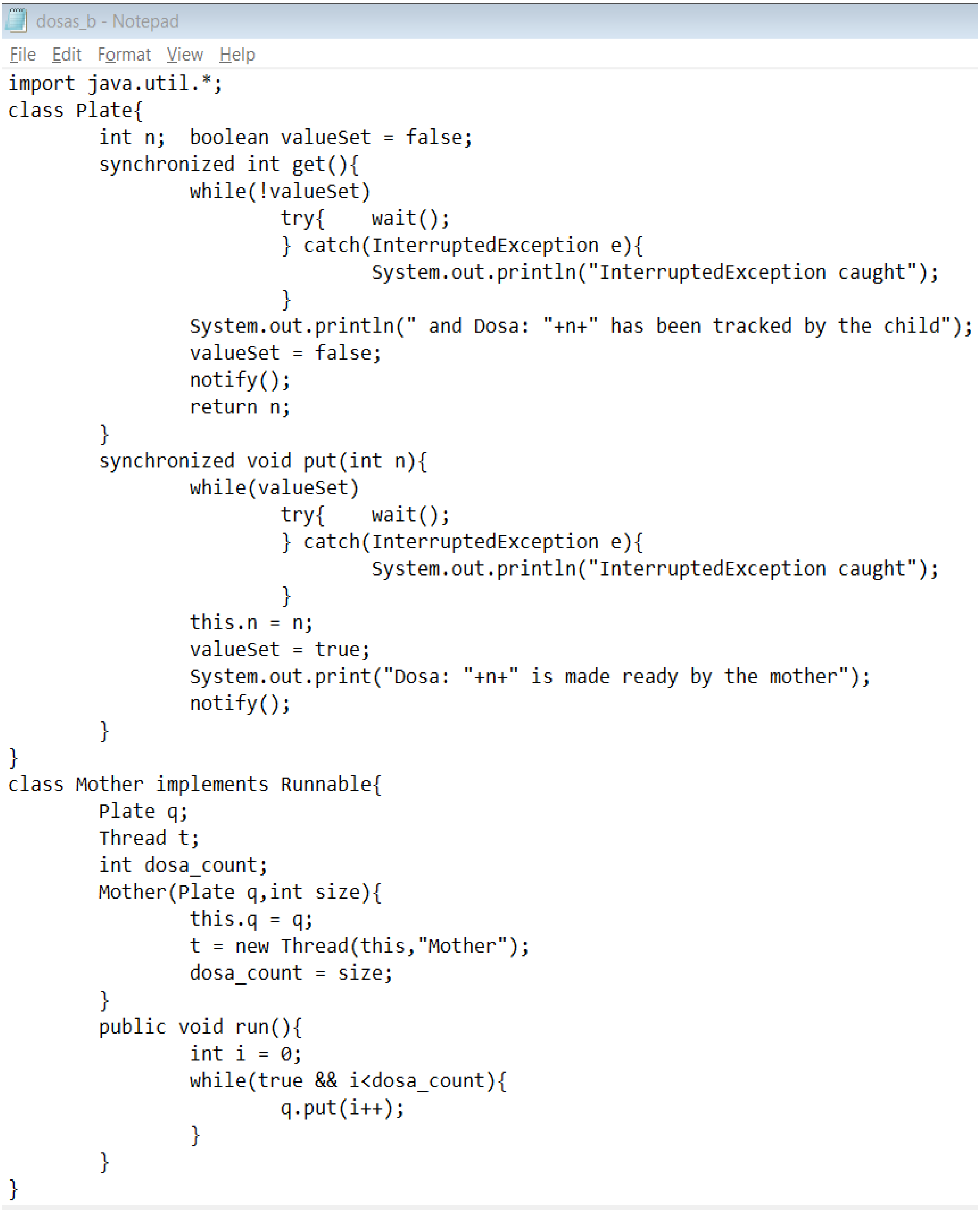
```
class Child extends Thread {
        Plate q;
        Child(Plate q){
                this.q = q;
        }

        public void run(){

                synchronized (this.q.buffer){
                        try {
                                q.buffer.wait();
                        }
                        catch (Exception e) {
                                e.printStackTrace();
                        }

                        for (int i = 0; i < q.buffer.capacity(); i++) {
                                System.out.println("Child has tracked dosa -"+q.buffer.charAt(i) +
                                 " being ready....Total dosas ready in the plate is "+(i+1));
                        }
                        System.out.println("Child has tracked that all the dosas are ready");
                }
        }
}


class dosas_a {
        public static void main(String args[]){
                Scanner sc = new Scanner(System.in);
                System.out.println("Enter the maximum count of dosas mother is planning to make");

                Plate q = new Plate();
                Mother p = new Mother(sc.nextInt(),q);
                Child c = new Child(q);

                Thread t1 = new Thread(p);
                Thread t2 = new Thread(c);

                t2.start();
                t1.start();
        }
}
```

OUTPUT IN TERMINAL – Q1



```
Microsoft Windows [Version 10.0.18363.1082]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Srihari>cd Desktop

C:\Users\Srihari\Desktop>cd javaweek9-threads

C:\Users\Srihari\Desktop\javaweek9-threads>javac dosas_a.java

C:\Users\Srihari\Desktop\javaweek9-threads>java dosas_a
Enter the maximum count of dosas mother is planning to make
5
Mother has prepared dosa -0
Mother has prepared dosa -1
Mother has prepared dosa -2
Mother has prepared dosa -3
Mother has prepared dosa -4
All dosas prepared
Child has tracked dosa -0 being ready....Total dosas ready in the plate is 1
Child has tracked dosa -1 being ready....Total dosas ready in the plate is 2
Child has tracked dosa -2 being ready....Total dosas ready in the plate is 3
Child has tracked dosa -3 being ready....Total dosas ready in the plate is 4
Child has tracked dosa -4 being ready....Total dosas ready in the plate is 5
Child has tracked that all the dosas are ready

C:\Users\Srihari\Desktop\javaweek9-threads>
```

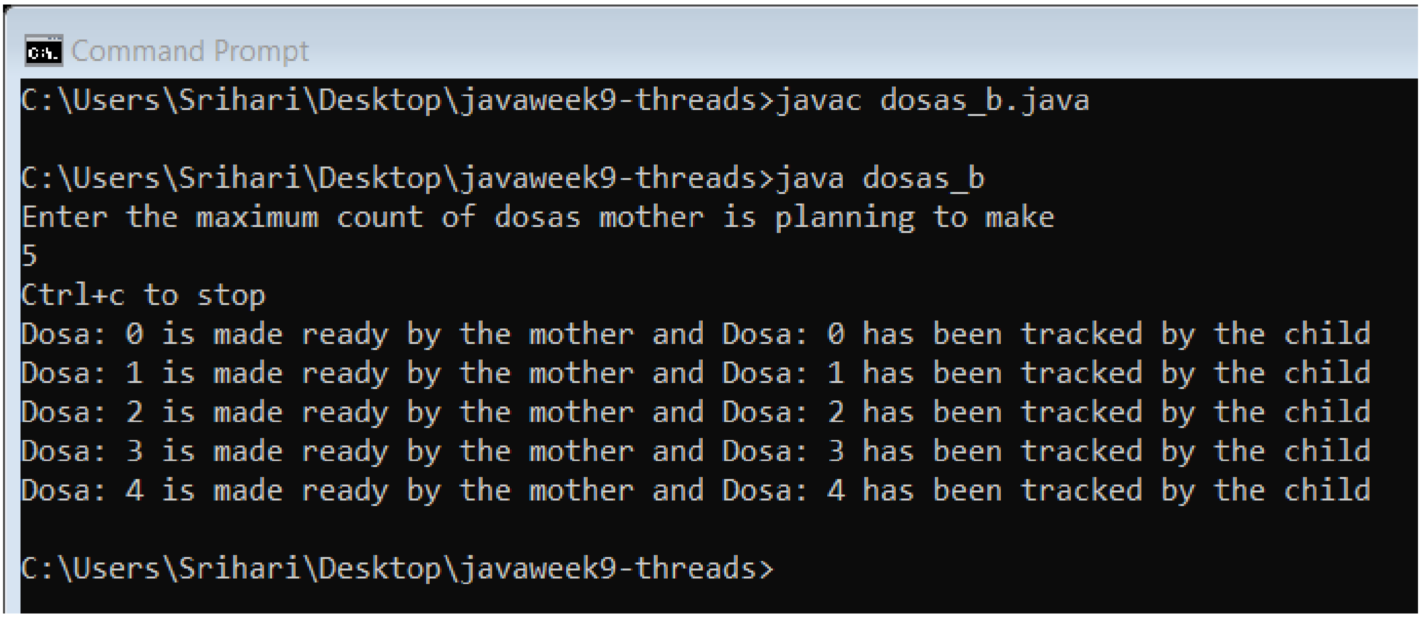
Q2. SYNCHRONIZATION IN THREADS



```
import java.util.*;
class Plate{
        int n;  boolean valueSet = false;
        synchronized int get(){
                while(!valueSet)
                        try{    wait();
                        } catch(InterruptedException e){
                                System.out.println("InterruptedException caught");
                        }
                System.out.println(" and Dosa: "+n+" has been tracked by the child");
                valueSet = false;
                notify();
                return n;
        }
        synchronized void put(int n){
                while(valueSet)
                        try{    wait();
                        } catch(InterruptedException e){
                                System.out.println("InterruptedException caught");
                        }
                this.n = n;
                valueSet = true;
                System.out.print("Dosa: "+n+" is made ready by the mother");
                notify();
        }
}
class Mother implements Runnable{
        Plate q;
        Thread t;
        int dosa_count;
        Mother(Plate q,int size){
                this.q = q;
                t = new Thread(this,"Mother");
                dosa_count = size;
        }
        public void run(){
                int i = 0;
                while(true && i<dosa_count){
                        q.put(i++);
                }
        }
}
```

```java
class Child implements Runnable{
        Plate q;
        Thread t;

        Child(Plate q){
                this.q = q;
                t = new Thread(this,"Child");
        }

        public void run(){
                while(true){
                        q.get();
                }
        }
}


class dosas_b{
        public static void main(String[] args){

                Scanner sc = new Scanner(System.in);
                System.out.println("Enter the maximum count"
                +" of dosas mother is planning to make");
                int size = sc.nextInt();
                Plate q = new Plate();
                Mother p = new Mother(q,size);
                Child c = new Child(q);

                p.t.start();
                c.t.start();
                System.out.println("Ctrl+c to stop");
        }
}
```

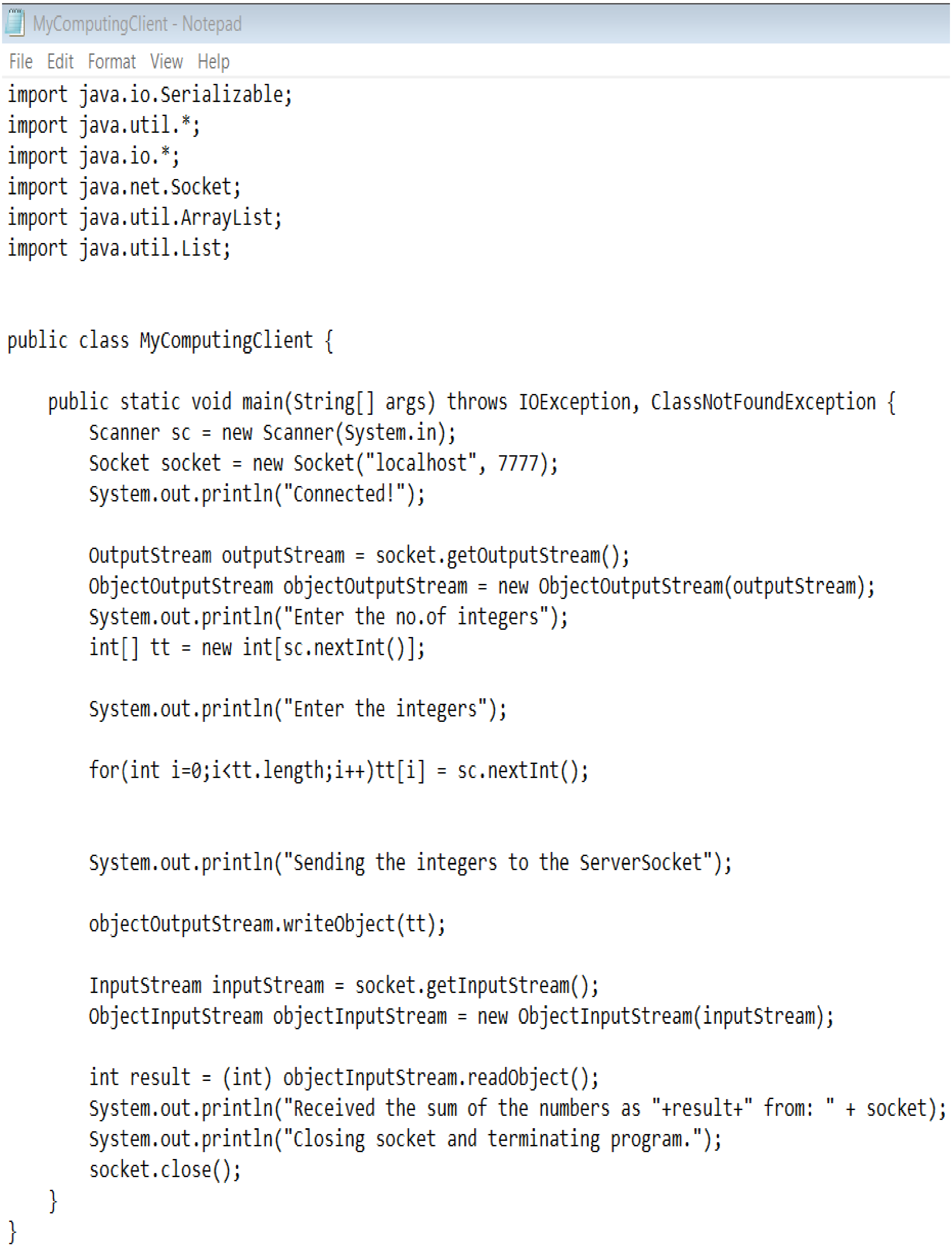
OUTPUT IN TERMINAL – Q2



```
C:\Users\Srihari\Desktop\javaweek9-threads>javac dosas_b.java

C:\Users\Srihari\Desktop\javaweek9-threads>java dosas_b
Enter the maximum count of dosas mother is planning to make
5
Ctrl+c to stop
Dosa: 0 is made ready by the mother and Dosa: 0 has been tracked by the child
Dosa: 1 is made ready by the mother and Dosa: 1 has been tracked by the child
Dosa: 2 is made ready by the mother and Dosa: 2 has been tracked by the child
Dosa: 3 is made ready by the mother and Dosa: 3 has been tracked by the child
Dosa: 4 is made ready by the mother and Dosa: 4 has been tracked by the child

C:\Users\Srihari\Desktop\javaweek9-threads>
```

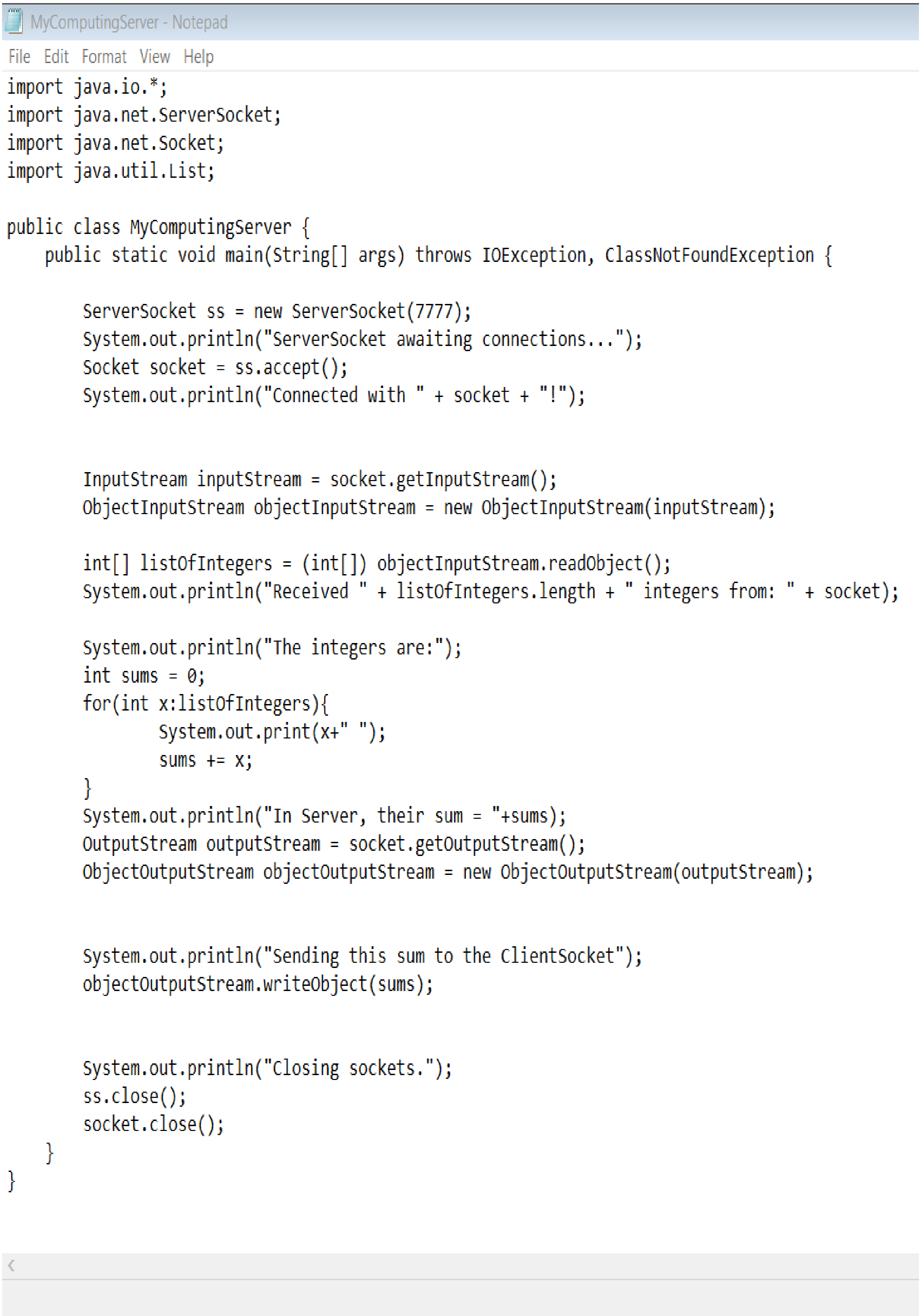
Q3. NETWORKING IN JAVA

MyComputingClient.java



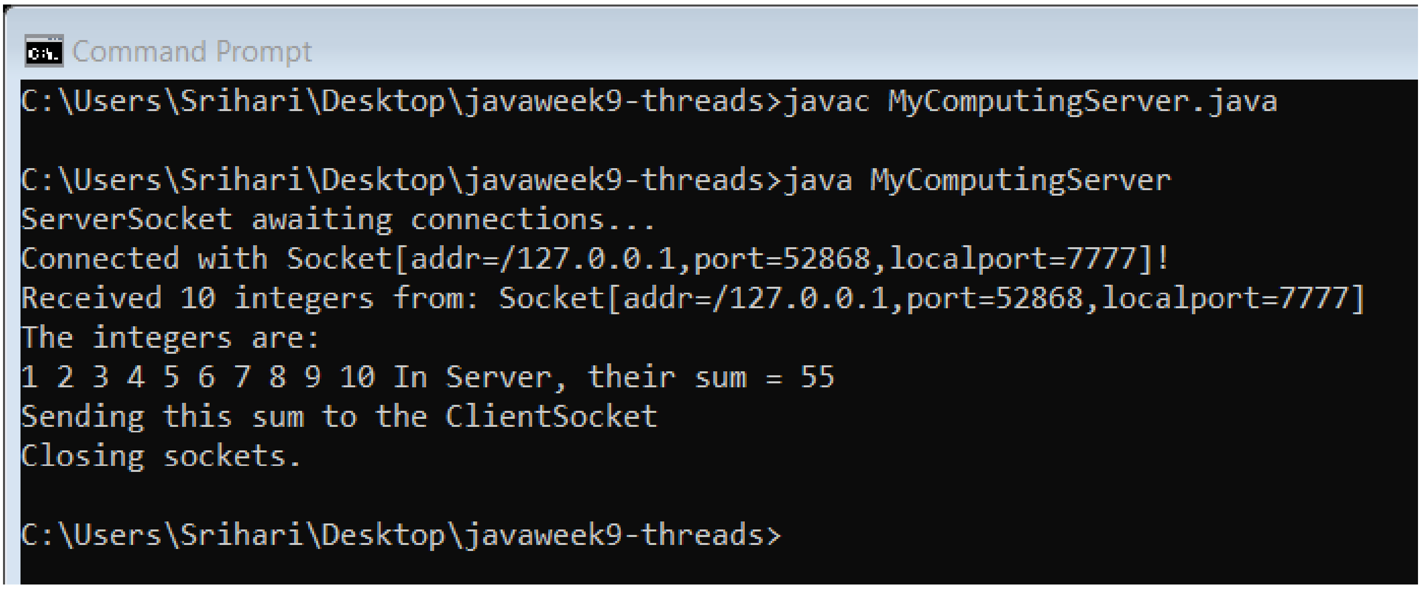
```java
import java.io.Serializable;
import java.util.*;
import java.io.*;
import java.net.Socket;
import java.util.ArrayList;
import java.util.List;


public class MyComputingClient {

    public static void main(String[] args) throws IOException, ClassNotFoundException {
        Scanner sc = new Scanner(System.in);
        Socket socket = new Socket("localhost", 7777);
        System.out.println("Connected!");

        OutputStream outputStream = socket.getOutputStream();
        ObjectOutputStream objectOutputStream = new ObjectOutputStream(outputStream);
        System.out.println("Enter the no.of integers");
        int[] tt = new int[sc.nextInt()];

        System.out.println("Enter the integers");

        for(int i=0;i<tt.length;i++)tt[i] = sc.nextInt();


        System.out.println("Sending the integers to the ServerSocket");

        objectOutputStream.writeObject(tt);

        InputStream inputStream = socket.getInputStream();
        ObjectInputStream objectInputStream = new ObjectInputStream(inputStream);

        int result = (int) objectInputStream.readObject();
        System.out.println("Received the sum of the numbers as "+result+" from: " + socket);
        System.out.println("Closing socket and terminating program.");
        socket.close();
    }
}
```

MyComputingServer.java



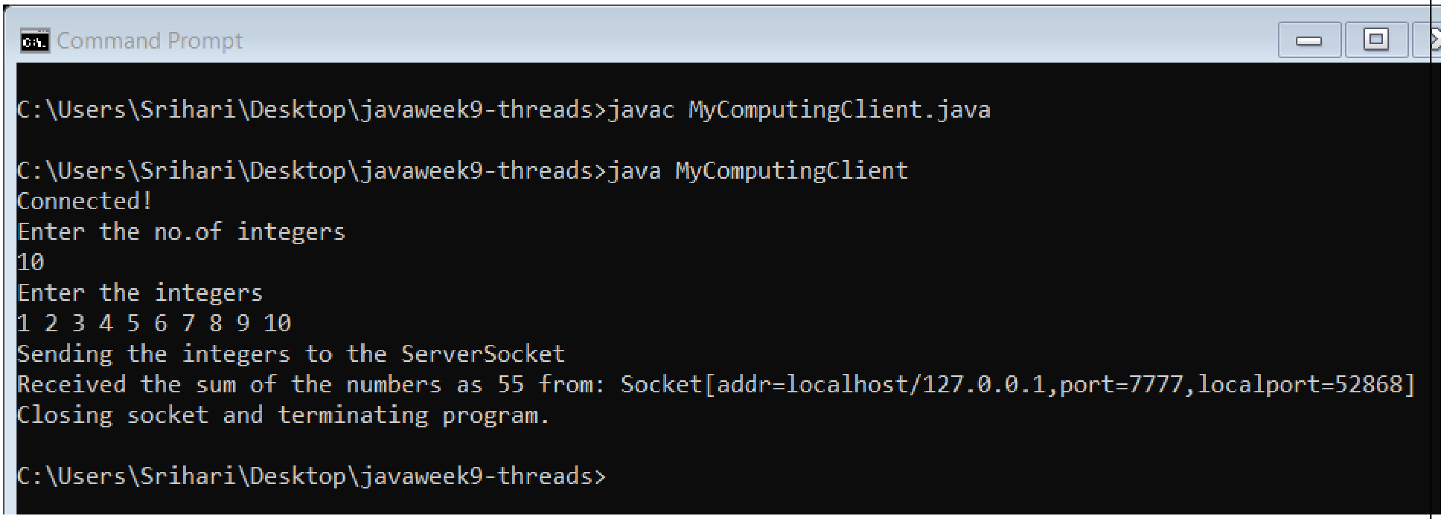
```java
import java.io.*;
import java.net.ServerSocket;
import java.net.Socket;
import java.util.List;

public class MyComputingServer {
    public static void main(String[] args) throws IOException, ClassNotFoundException {

        ServerSocket ss = new ServerSocket(7777);
        System.out.println("ServerSocket awaiting connections...");
        Socket socket = ss.accept();
        System.out.println("Connected with " + socket + "!");


        InputStream inputStream = socket.getInputStream();
        ObjectInputStream objectInputStream = new ObjectInputStream(inputStream);

        int[] listOfIntegers = (int[]) objectInputStream.readObject();
        System.out.println("Received " + listOfIntegers.length + " integers from: " + socket);

        System.out.println("The integers are:");
        int sums = 0;
        for(int x:listOfIntegers){
                System.out.print(x+" ");
                sums += x;
        }
        System.out.println("In Server, their sum = "+sums);
        OutputStream outputStream = socket.getOutputStream();
        ObjectOutputStream objectOutputStream = new ObjectOutputStream(outputStream);


        System.out.println("Sending this sum to the ClientSocket");
        objectOutputStream.writeObject(sums);


        System.out.println("Closing sockets.");
        ss.close();
        socket.close();
    }
}
```

MyComputingServer – OUTPUT in Terminal



```
C:\Users\Srihari\Desktop\javaweek9-threads>javac MyComputingServer.java

C:\Users\Srihari\Desktop\javaweek9-threads>java MyComputingServer
ServerSocket awaiting connections...
Connected with Socket[addr=/127.0.0.1,port=52868,localport=7777]!
Received 10 integers from: Socket[addr=/127.0.0.1,port=52868,localport=7777]
The integers are:
1 2 3 4 5 6 7 8 9 10 In Server, their sum = 55
Sending this sum to the ClientSocket
Closing sockets.

C:\Users\Srihari\Desktop\javaweek9-threads>
```

MyComputingClient – OUTPUT in Terminal



```
C:\Users\Srihari\Desktop\javaweek9-threads>javac MyComputingClient.java

C:\Users\Srihari\Desktop\javaweek9-threads>java MyComputingClient
Connected!
Enter the no.of integers
10
Enter the integers
1 2 3 4 5 6 7 8 9 10
Sending the integers to the ServerSocket
Received the sum of the numbers as 55 from: Socket[addr=localhost/127.0.0.1,port=7777,localport=52868]
Closing socket and terminating program.

C:\Users\Srihari\Desktop\javaweek9-threads>
```