Echo:

Host Part:

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 */

package com.mycompany.echohost1;

/**
 *
 * @author ThinkPad
 */
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 */

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

import java.io.IOException;
import java.util.logging.Level;
import java.util.logging.Logger;

/**
 *
 * @author ThinkPad
 */
public class EchoHost1 {

    public static void main(String[] args) throws IOException {
        int port= 6789;
        try(
            ServerSocket serversocket= new ServerSocket(port)
              )
        {
           System.out.println("Server is lisneting!!!");
           while(true)
           {
              Socket socket= serversocket.accept();
              System.out.println("Connected");
              new EchoHandler(socket).start();
           }
        }
```

```
    }
}
class EchoHandler extends Thread
{
    private final Socket socket;
    EchoHandler(Socket socket) {
        this.socket= socket;
    }
    @Override
    public void run()
    {
        try(
            BufferedReader reader= new BufferedReader(new InputStreamReader(socket.getInputStream()));
            PrintWriter writer= new PrintWriter(socket.getOutputStream(),true)
                )
        {
                String message= reader.readLine();
                System.out.println("Receive: "+ message);
                writer.println("Echo: "+message.toUpperCase());

        } catch (IOException ex) {
            Logger.getLogger(EchoHandler.class.getName()).log(Level.SEVERE, null, ex);
        }
    }

}
```

Client Part:

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 */

package com.mycompany.echoclient;

import java.io.*;
import java.net.*;
/**
 *
 * @author ThinkPad
 */
public class EchoClient {

    public static void main(String[] args) throws IOException {
        String hostname= "localhost";
```

```
        int port= 6789;

        try(
            Socket socket= new Socket(hostname, port);
            PrintWriter write= new PrintWriter(socket.getOutputStream(), true);
            BufferedReader reader= new BufferedReader(new
InputStreamReader(socket.getInputStream()));
            BufferedReader consoleReader= new BufferedReader(new InputStreamReader(System.in))
            ){
          String userInput;
          while(true)
          {
            userInput = consoleReader.readLine();
            write.println(userInput);
            String response= reader.readLine();
            System.out.println(response);
          }
        }
    }
}
```

Math_Calculator:

Host Part:

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 */

package com.mycompany.mathhost;

/**
 *
 * @author ThinkPad Khairul Islam Robi
 */
import java.io.*;
import java.net.*;
import java.util.logging.Level;
import java.util.logging.Logger;
public class MathHost {

    public static void main(String[] args) throws IOException {
        int port= 6789;
        try(ServerSocket serversocket= new ServerSocket(port))
        {
```

```
            System.out.println("Server is listening on port: "+ port);
            while(true)
            {
               Socket socket= serversocket.accept();
               System.out.println("New Client Connected!!!");
               new MathHandler(socket).start();
            }
         }
      }
}
class MathHandler extends Thread {
   private final Socket socket;
   public MathHandler(Socket socket) {
      this.socket= socket;
   }

   @Override
   public void run()
   {
      try(
         PrintWriter writer= new PrintWriter(socket.getOutputStream(),true);
         BufferedReader reader= new BufferedReader(new InputStreamReader(socket.getInputStream()));
            )
      {
         while(true){
         String number= reader.readLine();
         System.out.println("Received: "+number);
         String response= Calculator(number);
         writer.println(response);
         }
      } catch (IOException ex) {
         Logger.getLogger(MathHandler.class.getName()).log(Level.SEVERE, null, ex);
      }
   }


   private String Calculator(String number) {
      String[] parts= number.split(" ");
      String operator= parts[1];
      double number1, number2;
      try{
         number1= Double.parseDouble(parts[0]);
         number2= Double.parseDouble(parts[2]);
      }
```

```
        catch(NumberFormatException e)
        {
            return "Error";
        }

        double result;
        switch(operator)
        {
            case "+" -> result= number1 + number2;
            case "-" -> result= number1 - number2;
            case "*" -> result= number1 * number2;
            case "/" -> result= number1 / number2;
            default -> {
                return "Error: Unknown Operator";
            }
        }
        return "Result: "+result;
    }

}
```

Client Part:

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 */

package com.mycompany.mathclient;

/**
 *
 * @author ThinkPad Khairul Islam Robi
 */
import java.io.*;
import java.net.*;
public class MathClient {

    public static void main(String[] args) throws IOException {
        while(true){
        int port= 6789;
        String hostname= "localhost";
        try(
            Socket socket= new Socket(hostname, port);
            PrintWriter writer= new PrintWriter(socket.getOutputStream(), true);
            BufferedReader reader= new BufferedReader(new InputStreamReader(socket.getInputStream()));
```

```
        BufferedReader consoleReader= new BufferedReader(new InputStreamReader(System.in))
            )
      {
        String userInput;
        while((userInput=consoleReader.readLine())!= null)
        {
          System.out.println("Enter number_1 symbol number_2: ");
          writer.println(userInput);
          String response= reader.readLine();
          System.out.println(response);

        }
      }
   }
   }
}
```

Largest_Number:

Host Part:

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 */

package com.mycompany.largestnumberhost;

/**
 *
 * @author ThinkPad
 */
import java.io.*;
import java.net.*;
import java.util.logging.Level;
import java.util.logging.Logger;
public class LargestNumberHost {

   public static void main(String[] args) throws IOException {
     int port = 6789;
     try(ServerSocket serversocket = new ServerSocket(port))
     {
        while(true)
        {
          Socket socket= serversocket.accept();
          System.out.println("Listening on port "+port);
```

```
            new ClientHandler(socket).start();
        }
     }
   }
}
class ClientHandler extends Thread {
   private final Socket socket;
   public ClientHandler(Socket socket) {
      this.socket= socket;
   }
   @Override
   public void run()
   {
      try(BufferedReader reader = new BufferedReader(new
InputStreamReader(socket.getInputStream()));
          PrintWriter writer = new PrintWriter(socket.getOutputStream(), true)
          )
      {
        while(true){
        String numberString = reader.readLine();
        System.out.println(numberString);
        String response = Largest(numberString);
        writer.println(response);
        }
      } catch (IOException ex) {
         Logger.getLogger(ClientHandler.class.getName()).log(Level.SEVERE, null, ex);
      }
   }

   private String Largest(String numberString) {
      String[] number = numberString.split(" ");
      int Max= Integer.parseInt(number[0]);
      for(String num: number)
      {
         int m = Integer.parseInt(num);
         if(m > Max)
         {
            Max = m;
         }
      }
      return "Largest Number is " + Max;

   }
}
```

Client Part:

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 */

package com.mycompany.largestnumberclient;

/**
 *
 * @author ThinkPad
 */
import java.io.*;
import java.net.*;
import java.util.logging.Level;
import java.util.logging.Logger;
public class LargestNumberClient {

   public static void main(String[] args) {
      int port = 6789;
      String hostname = "localhost";

      try( Socket socket = new Socket(hostname, port);
            BufferedReader reader = new BufferedReader(new
InputStreamReader(socket.getInputStream()));
               PrintWriter writer = new PrintWriter(socket.getOutputStream(), true);
                  BufferedReader consoleReader = new BufferedReader(new
InputStreamReader(System.in))
            )
      {
        String userInput;
        System.out.print("Enter Series of Number: ");
        while((userInput = consoleReader.readLine())!=null){
            writer.println(userInput);
            String response = reader.readLine();
            System.out.println(response);
        }
      } catch (IOException ex) {
         Logger.getLogger(LargestNumberClient.class.getName()).log(Level.SEVERE, null, ex);
      }
   }

}
```

IP_Address:

Host Part:

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 */

package com.mycompany.lab_3_host;

/**
 *
 * @author ThinkPad
 */
import java.io.*;
import java.net.*;
import java.util.logging.Level;
import java.util.logging.Logger;
public class Lab_3_Host {

    public static void main(String[] args) throws IOException {
        int port = 9876;
        try(ServerSocket serversocket = new ServerSocket(port))
        {
            System.out.println("Server is Running....");
            while(true)
            {
                Socket socket = serversocket.accept();
                System.out.println("Server is listening on port " +port);
                new IP_Handler(socket).start();
            }
        }
    }
}
class IP_Handler extends Thread {
    private final Socket socket;
        public IP_Handler(Socket socket) {
            this.socket = socket;
        }
    @Override
    public void run()
    {
```

```
        try(BufferedReader reader = new BufferedReader(new
InputStreamReader(socket.getInputStream()));
                PrintWriter writer = new PrintWriter(socket.getOutputStream(), true)
                )
        {
           while(true)
           {
              String request = reader.readLine();
              System.out.println("Received: "+request);
              String response = IP_Calculator(request);
              writer.println(response);
           }
        } catch (IOException ex) {
           Logger.getLogger(IP_Handler.class.getName()).log(Level.SEVERE, null, ex);
        }
     }
     private String IP_Calculator(String request) throws UnknownHostException
     {
        InetAddress inet = InetAddress.getByName(request);
        String ip = inet.getHostAddress();
        String[] parts = ip.split("\\.");
        int f = Integer.parseInt(parts[0]);
        if(f>=1 && f<128)
        {
           return "Class A and the IP is "+ip;
        }
        else if(f>=128 && f<192)
        {
           return "Class B and the IP is "+ip;
        }
        else if(f>=192 && f<224)
        {
           return "Class C and the IP is "+ip;
        }
        else if(f>=224 && f<240)
        {
           return "Class D and the IP is "+ip;
        }
        else if(f>=240 && f<=255)
        {
           return "Class E and the IP is "+ip;
        }
        return "Invalid IP Address "+ip;
```

```
    }
}
```

Client Part:

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 */

package com.mycompany.lab_3_client;

/**
 *
 * @author ThinkPad
 */
import java.io.*;
import java.net.*;
public class Lab_3_Client {

    public static void main(String[] args) throws IOException {
        int port = 9876;
        String hostname = "localhost";
        try(Socket socket = new Socket(hostname, port);
                BufferedReader reader = new BufferedReader(new
InputStreamReader(socket.getInputStream()));
                PrintWriter writer = new PrintWriter(socket.getOutputStream(), true);
                BufferedReader consolereader = new BufferedReader(new InputStreamReader(System.in))
                )
        {
            while(true)
            {
                System.out.println("Enter IP Address or exit: ");
                String userInput = consolereader.readLine();
                if(userInput.equalsIgnoreCase("exit"))
                {
                    System.out.println("Thanks.....");
                    break;
                }
                writer.println(userInput);
                String response = reader.readLine();
                System.out.println(response);
            }
        }
    }
}
```