NAME	JACOB JOHN
REGISTER NO.	16BCE2205
E-MAIL	jacob.john2016@vitstudent.ac.in
COURSE	Java Programming

LAB ASSESSMENT #3

All labs can also be found at:

https://github.com/jacobjohn2016/Java-Programming

Applets

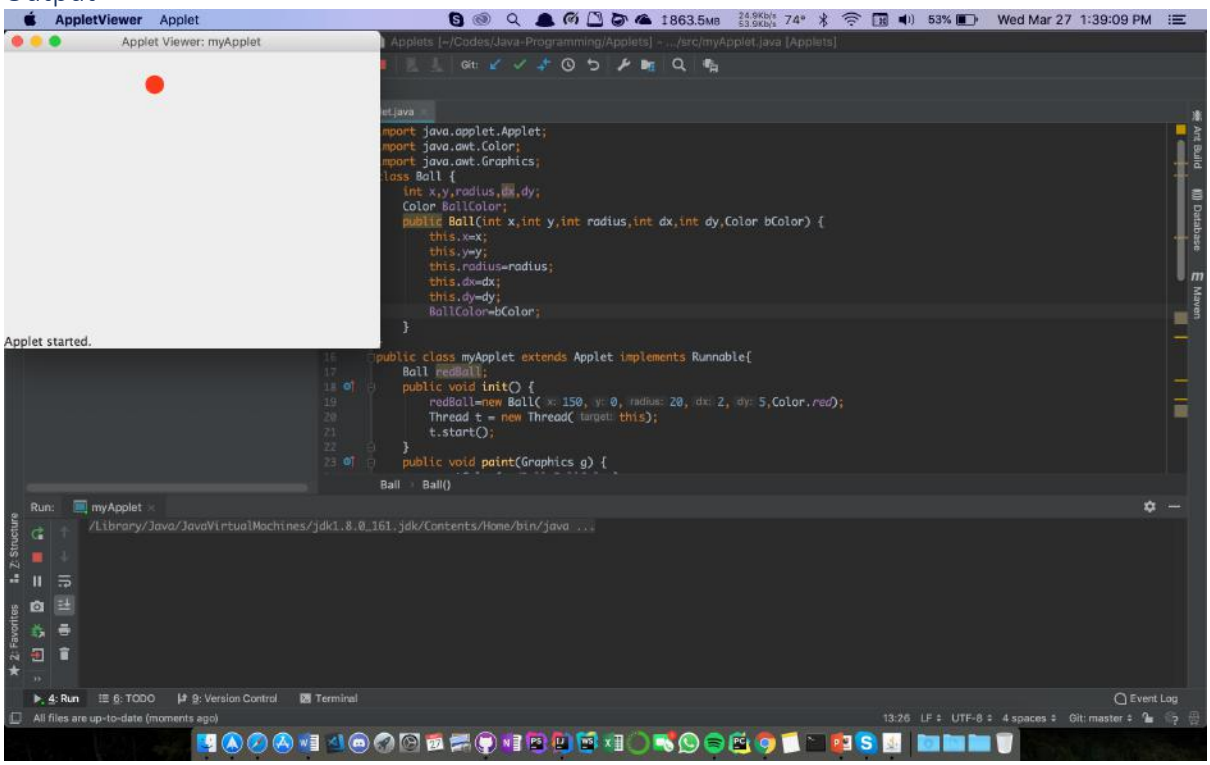
Draw a ball, filled with default color. Move the ball from top to bottom of the window continuously with its color changed for every one second. The new color of the ball for the next second should be obtained by adding 20 to the current value of Red component, for the second time by adding 20 to the blue component, and for the third time by adding 20 to the blue component, till all reach the final limit 225, after which the process should be repeated with the default color.

Code

```
import
java.applet.Applet;
                        import java.awt.Color;
                        import java.awt.Graphics;
                        class Ball {
                            int x,y,radius,dx,dy;
                            Color BallColor;
                            public Ball(int x,int y,int radius,int dx,int dy,Color
                        bColor) {
                                this.x=x;
                                this.y=y;
                                this.radius=radius;
                                this.dx=dx;
                                this.dy=dy;
                                BallColor=bColor;
                            }
                        }
                        public class myApplet extends Applet implements Runnable{
                            Ball redBall;
                            public void init() {
                                redBall=new Ball(150,0,20,2,5,Color.red);
```

```
                Thread t = new Thread(this);
                t.start();
            }
            public void paint(Graphics g) {
                g.setColor(redBall.BallColor);
                g.fillOval(redBall.x, redBall.y, redBall.radius,
        redBall.radius);
            }
            public void run() {
                try {
                    Color c = redBall.BallColor;
                    redBall.BallColor = new Color(c.getRed(),
        c.getBlue()+20, c.getGreen());
                    redBall.y=redBall.y+redBall.dy;
                }
                catch(Exception e){}
                while(redBall.BallColor.getBlue()!=255) {
                    try {
                        displacementOperation(redBall);
                        Thread.sleep(1000);
                        repaint();
                    }
                    catch(Exception e){}
                }
            }
            public void displacementOperation(Ball ball) {
                Color c = ball.BallColor;
                ball.BallColor = new Color(c.getRed(), c.getBlue()+20,
        c.getGreen());
                ball.y=ball.y+ball.dy;
            }
        }
```

Output


AppletViewer Applet
Wed Mar 27 1:39:09 PM
Applet Viewer: myApplet
Applet started.
import java.applet.Applet;
import java.awt.Color;
import java.awt.Graphics;
class Ball {
int x,y,radius,dx,dy;
Color BallColor;
public Ball(int x,int y,int radius,int dx,int dy,Color bColor) {
this.x=x;
this.y=y;
this.radius=radius;
this.dx=dx;
this.dy=dy;
BallColor=bColor;
}
public class myApplet extends Applet implements Runnable{
Ball redBall;
public void init() {
redBall=new Ball(x: 150, y: 0, radius: 20, dx: 2, dy: 5,Color.red);
Thread t = new Thread(target: this);
t.start();
}
public void paint(Graphics g) {
Ball › Ball()
Run: myApplet
/Library/Java/JavaVirtualMachines/jdk1.8.0_161.jdk/Contents/Home/bin/java ...
4: Run
6: TODO
9: Version Control
Terminal
Event Log
All files are up-to-date (moments ago)
13:26 LF UTF-8 4 spaces Git: master

Java Network Programming

Develop a UDP based client-server application to notify the client about the integrity of data sent from its side. Use checksum to do this.

Server

```
// Java
program to
illustrate
Server
side
```

```
// Implementation using DatagramSocket
import java.io.IOException;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.net.SocketException;

public class udpBaseServer_2
{
    public static void main(String[] args) throws IOException
    {
        // Step 1 : Create a socket to listen at port 1234
        DatagramSocket ds = new DatagramSocket(1234);
        byte[] receive = new byte[65535];

        DatagramPacket DpReceive = null;

        // Step 2 : create a DatgramPacket to receive the data.
        DpReceive = new DatagramPacket(receive, receive.length);

        // Step 3 : revieve the data in byte buffer.
        ds.receive(DpReceive);

        System.out.println("Client:-" + data(receive));
        String ls = data(receive).toString();
        int l = Integer.valueOf(ls);

        int i, nob, sum = 0, chk_sum;

        // Initializes the arrays based on data length received
        int c_data[] = new int[l];
        int data[] = new int[l];

        System.out.println("Data received (along with checksum) is");
```

```
                for(i = 0; i< data.length; i++)
                {
                    // Step 2 : create a DatgramPacket to receive the data.
                    DpReceive = new DatagramPacket(receive, receive.length);
                    // Step 3 : revieve the data in byte buffer.
                    ds.receive(DpReceive);
                    System.out.println("Client : " + data(receive));
                    ls = data(receive).toString();
                    // Reading the data being sent one by one
                    data[i] = Integer.valueOf(ls);

                    // Complementing the data being received
                    nob = (int)(Math.floor(Math.log(data[i]) / Math.log(2))) +
1;
                    c_data[i] = ((1 << nob) - 1) ^ data[i];

                    // Adding the complemented data
                    sum += c_data[i];
                }
                System.out.println("Sum (in ones complement) is : "+sum);

                // Complementing the sum
                nob = (int)(Math.floor(Math.log(sum) / Math.log(2))) + 1;
                sum = ((1 << nob) - 1) ^ sum;
                System.out.println("Calculated Checksum is : "+sum);

                if(sum == 0){
                    System.out.println("Success!");
                }
                else{
                    System.out.println("Failure");
                }
            }

            // A utility method to convert the byte array
            // data into a string representation.
            public static StringBuilder data(byte[] a)
            {
                if (a == null)
                    return null;
                StringBuilder ret = new StringBuilder();
                int i = 0;
                while (a[i] != 0)
                {
                    ret.append((char) a[i]);
                    i++;
```

```
                    }
                    return ret;
                }
            }
```

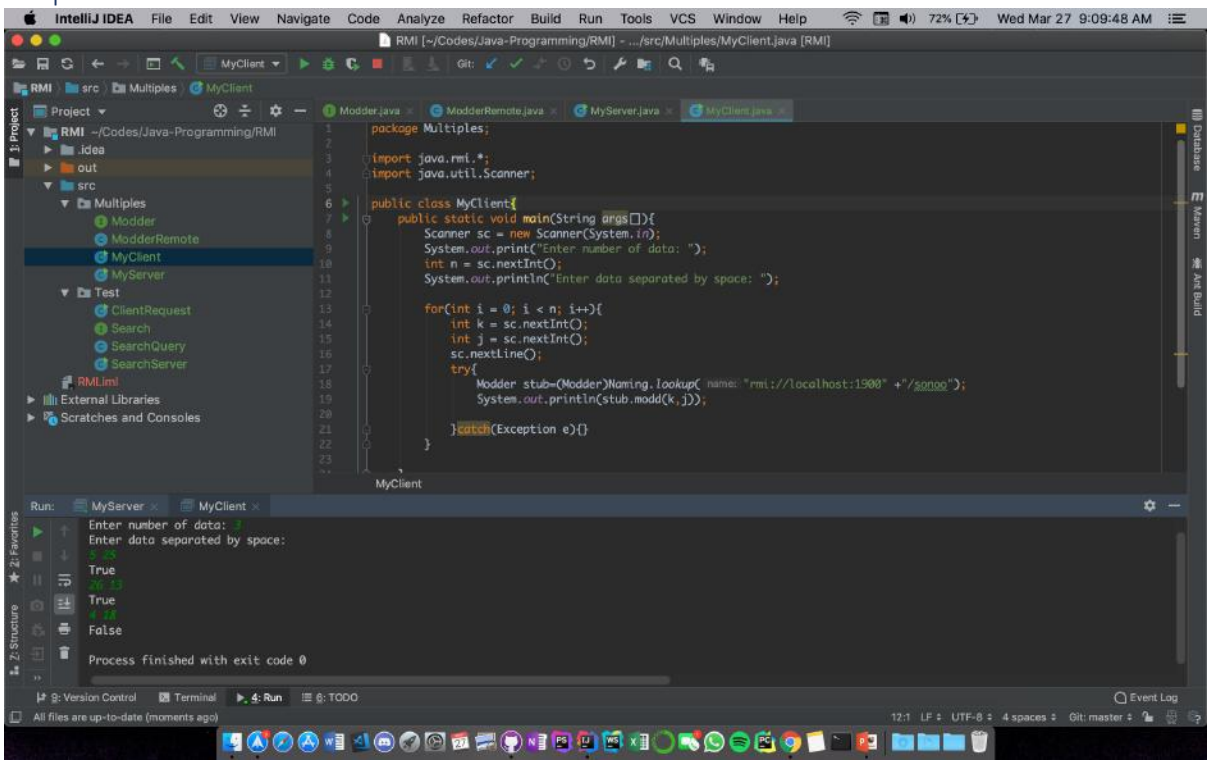
Client

```
// Java
program to
illustrate
Client
side
            // Implementation using DatagramSocket
            import java.io.IOException;
            import java.net.DatagramPacket;
            import java.net.DatagramSocket;
            import java.net.InetAddress;
            import java.util.Scanner;

            public class udpBaseClient_2
            {
                public static void main(String args[]) throws IOException
                {
                    // Setting maximum data length
                    int MAX = 100;
                    Scanner sc = new Scanner(System.in);

                    // Step 1:Create the socket object for
                    // carrying the data.
                    DatagramSocket ds = new DatagramSocket();

                    InetAddress ip = InetAddress.getLocalHost();
                    byte buf[] = null;

                    int i, l, sum = 0, nob;
                    System.out.print("Enter data length : ");
                    l = sc.nextInt();

                    // Array to hold the data being entered
                    int data[] = new int[MAX];

                    // Array to hold the complement of each data
                    int c_data[] = new int[MAX];

                    System.out.println("Enter data to send : ");
```

```
            for (i = 0; i < l; i++)
            {
                data[i] = sc.nextInt();

                // Complementing the entered data
                // Here we find the number of bits required to represent
                // the data, like say 8 requires 1000, i.e 4 bits
                nob = (int)(Math.floor(Math.log(data[i]) / Math.log(2))) +
1;

                // Here we do a XOR of the data with the number 2^n -1,
                // where n is the nob calculated in previous step
                c_data[i] = ((1 << nob) - 1) ^ data[i];

                // Adding the complemented data and storing in sum
                sum += c_data[i];
            }
            // The sum(i.e checksum) is also sent along with the data
            data[i] = sum;
            l += 1;

            System.out.println("Checksum Calculated is : " + sum);
            System.out.println("Data being sent along with Checksum...");

            // Sends the data length to receiver
            // convert the input into the byte array.
            String temp = Integer.toString(l);
            buf = temp.getBytes();

            // Step 2 : Create the datagramPacket for sending
            // the data.
            DatagramPacket DpSend = new DatagramPacket(buf, buf.length, ip,
1234);

            // Step 3 : invoke the send call to actually send
            // the data.
            ds.send(DpSend);

            // Sends the data one by one to receiver
            for (int j = 0; j < l; j++){
                temp = Integer.toString(data[j]);
                buf = temp.getBytes();
                DpSend = new DatagramPacket(buf, buf.length, ip, 1234);
                ds.send(DpSend);
            }
```

```
                }
            }
```

Output


IntelliJ IDEA File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
RMI [~/Codes/Java-Programming/RMI] - .../src/Multiples/MyClient.java [RMI]
Project
RMI ~/Codes/Java-Programming/RMI
.idea
out
src
Multiples
Modder
ModderRemote
MyClient
MyServer
Test
ClientRequest
Search
SearchQuery
SearchServer
RMI.iml
External Libraries
Scratches and Consoles
Modder.java
ModderRemote.java
MyServer.java
MyClient.java
package Multiples;
import java.rmi.*;
import java.util.Scanner;
public class MyClient{
public static void main(String args[]){
Scanner sc = new Scanner(System.in);
System.out.print("Enter number of data: ");
int n = sc.nextInt();
System.out.println("Enter data separated by space: ");
for(int i = 0; i < n; i++){
int k = sc.nextInt();
int j = sc.nextInt();
sc.nextLine();
try{
Modder stub=(Modder)Naming.lookup(name: "rmi://localhost:1900" +"/sonoo");
System.out.println(stub.modd(k,j));
}catch(Exception e){}
}
}
MyClient
Run: MyServer MyClient
Enter number of data: 3
Enter data separated by space:
5 15
True
26 13
True
4 18
False
Process finished with exit code 0
9: Version Control Terminal 4: Run 6: TODO
Event Log
All files are up-to-date (moments ago)
12:1 LF UTF-8 4 spaces Git: master

RMI

Develop an RMI application to invoke a remote method that takes two numbers and returns true if one number is an exact multiple of the other and false otherwise.

Remote interface

```
package
Multiples;

            import java.rmi.*;
            public interface Modder extends Remote{
                public String modd(int x,int y)throws RemoteException;
            }
```

Implementation

```
package
Multiples;

            import java.rmi.*;
            import java.rmi.server.*;
            public class ModderRemote extends UnicastRemoteObject implements Modder{
                ModderRemote()throws RemoteException{
                    super();
                }
                public String modd(int x,int y){
                    if(x%y == 0 || y%x == 0){
                        return "True";
                    }
                    else
                        return "False";
                }
```

Server

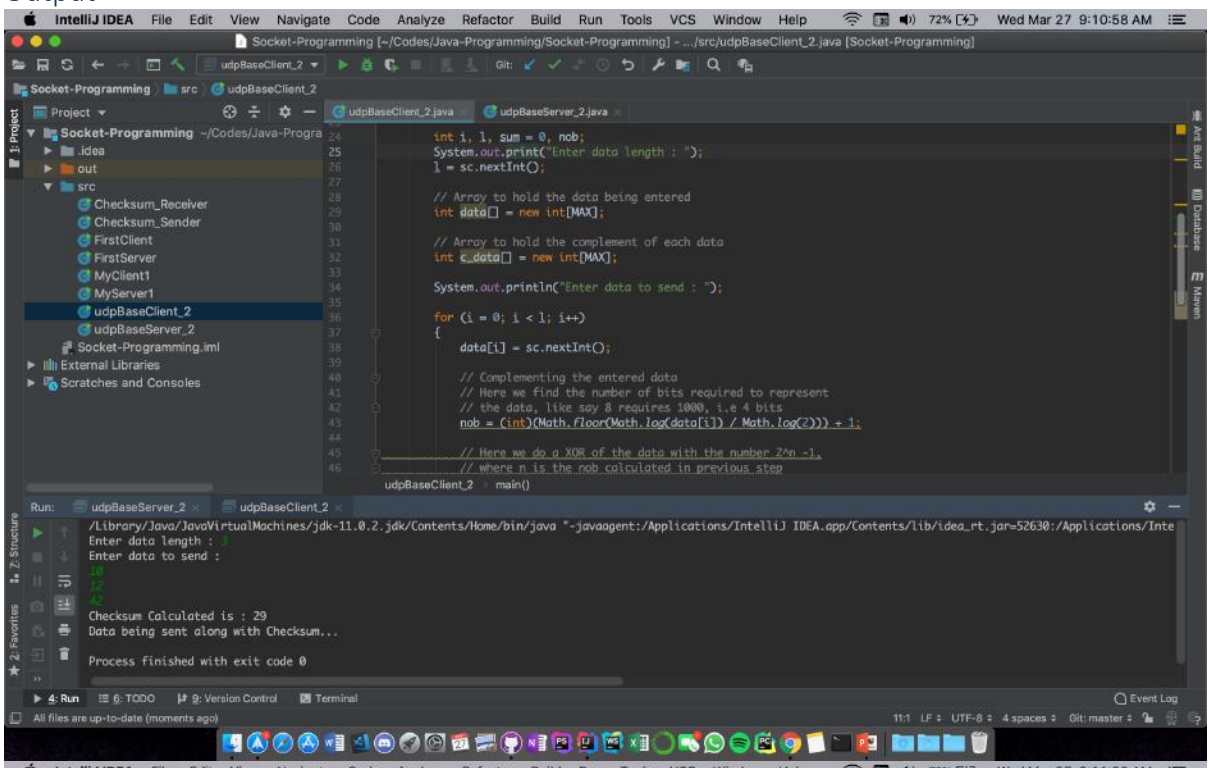
```
package
Multiples;

            import java.rmi.*;
            import java.rmi.registry.*;
            public class MyServer{
                public static void main(String args[]){
                    try{
                        Modder stub=new ModderRemote();

                        // rmiregistry within the server JVM with
                        // port number 1900
                        LocateRegistry.createRegistry(1900);
```

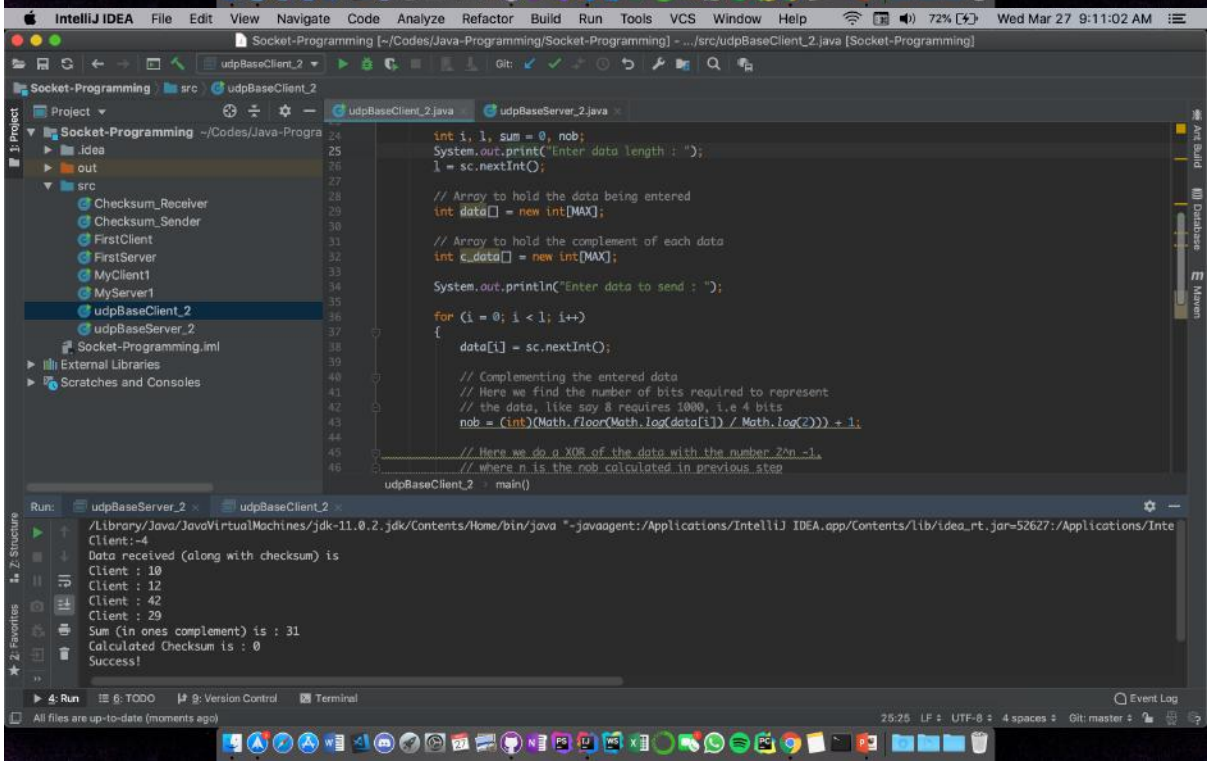
```
                    // Binds the remote object by the name
                    // sonoo
                    Naming.rebind("rmi://localhost:1900" + "/sonoo",stub);
                }catch(Exception e){System.out.println(e);}
            }
        }
```

Client

```
package
Multiples;

        import java.rmi.*;
        import java.util.Scanner;

        public class MyClient{
            public static void main(String args[]){
                Scanner sc = new Scanner(System.in);
                System.out.print("Enter number of data: ");
                int n = sc.nextInt();
                System.out.println("Enter data separated by space: ");

                for(int i = 0; i < n; i++){
                    int k = sc.nextInt();
                    int j = sc.nextInt();
                    sc.nextLine();
                    try{
                        Modder stub=(Modder)Naming.lookup("rmi://localhost:1900"
        +"/sonoo");
                        System.out.println(stub.modd(k,j));

                    }catch(Exception e){}
                }

            }
        }
```

Output


Socket-Programming [~/Codes/Java-Programming/Socket-Programming] - .../src/udpBaseClient_2.java [Socket-Programming]
udpBaseClient_2.java
udpBaseServer_2.java
int i, l, sum = 0, nob;
System.out.print("Enter data length : ");
l = sc.nextInt();
// Array to hold the data being entered
int data[] = new int[MAX];
// Array to hold the complement of each data
int c_data[] = new int[MAX];
System.out.println("Enter data to send : ");
for (i = 0; i < l; i++)
{
data[i] = sc.nextInt();
// Complementing the entered data
// Here we find the number of bits required to represent
// the data, like say 8 requires 1000, i.e 4 bits
nob = (int)(Math.floor(Math.log(data[i]) / Math.log(2))) + 1;
// Here we do a XOR of the data with the number 2^n -1,
// where n is the nob calculated in previous step
udpBaseClient_2 › main()
Run: udpBaseServer_2 udpBaseClient_2
/Library/Java/JavaVirtualMachines/jdk-11.0.2.jdk/Contents/Home/bin/java "-javaagent:/Applications/IntelliJ IDEA.app/Contents/lib/idea_rt.jar=52630:/Applications/Inte
Enter data length : 3
Enter data to send :
10
12
42
Checksum Calculated is : 29
Data being sent along with Checksum...
Process finished with exit code 0


Socket-Programming [~/Codes/Java-Programming/Socket-Programming] - .../src/udpBaseClient_2.java [Socket-Programming]
udpBaseClient_2.java
udpBaseServer_2.java
int i, l, sum = 0, nob;
System.out.print("Enter data length : ");
l = sc.nextInt();
// Array to hold the data being entered
int data[] = new int[MAX];
// Array to hold the complement of each data
int c_data[] = new int[MAX];
System.out.println("Enter data to send : ");
for (i = 0; i < l; i++)
{
data[i] = sc.nextInt();
// Complementing the entered data
// Here we find the number of bits required to represent
// the data, like say 8 requires 1000, i.e 4 bits
nob = (int)(Math.floor(Math.log(data[i]) / Math.log(2))) + 1;
// Here we do a XOR of the data with the number 2^n -1,
// where n is the nob calculated in previous step
udpBaseClient_2 › main()
Run: udpBaseServer_2 udpBaseClient_2
/Library/Java/JavaVirtualMachines/jdk-11.0.2.jdk/Contents/Home/bin/java "-javaagent:/Applications/IntelliJ IDEA.app/Contents/lib/idea_rt.jar=52627:/Applications/Inte
Client:-4
Data received (along with checksum) is
Client : 10
Client : 12
Client : 42
Client : 29
Sum (in ones complement) is : 31
Calculated Checksum is : 0
Success!

XML

```xml
<?xml
version="1.0"
encoding="UTF-
8"?>
                <web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                xmlns="http://xmlns.jcp.org/xml/ns/javaee"
                xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
                http://xmlns.jcp.org/xml/ns/javaee/web-app_4_0.xsd" id="WebApp_ID"
                version="4.0">
                  <display-name>Servlet-Programming</display-name>
                  <welcome-file-list>
                    <welcome-file>index.html</welcome-file>
                    <welcome-file>index.htm</welcome-file>
                    <welcome-file>index.jsp</welcome-file>
                    <welcome-file>default.html</welcome-file>
                    <welcome-file>default.htm</welcome-file>
                    <welcome-file>default.jsp</welcome-file>
                  </welcome-file-list>
                  <servlet>
                    <servlet-name>HelloForm</servlet-name>
                    <servlet-class>HelloForm</servlet-class>
                  </servlet>
                  <servlet-mapping>
                    <servlet-name>HelloForm</servlet-name>
                    <url-pattern>/HelloForm</url-pattern>
                  </servlet-mapping>
                  <servlet>
                      <servlet-name>ReadCookies</servlet-name>
                      <servlet-class>ReadCookies</servlet-class>
                  </servlet>
                  <servlet-mapping>
                      <servlet-name>ReadCookies</servlet-name>
                      <url-pattern>/ReadCookies</url-pattern>
                  </servlet-mapping>
                  <servlet>
                      <servlet-name>DeleteCookies</servlet-name>
                      <servlet-class>DeleteCookies</servlet-class>
                  </servlet>
                  <servlet-mapping>
                      <servlet-name>DeleteCookies</servlet-name>
                      <url-pattern>/DeleteCookies</url-pattern>
                  </servlet-mapping>
```

```
</web-app>
```

Hello.html

```
<html>
   <body>
      <form action = "HelloForm" method = "GET">
         First Name: <input type = "text" name = "first_name">
         <br />
         Last Name: <input type = "text" name = "last_name" />
         <br />
         <input type = "radio" name = "campus" value = "Chennai" checked/>
Chennai Campus
         <br />
         <input type = "radio" name = "campus" value = "Vellore"  /> Vellore
Campus
         <br />
         <input type = "submit" value = "Submit" />
      </form>
   </body>
</html>
```

HelloForm.java

```
// Import
required
java
libraries
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

// Extend HttpServlet class
public class HelloForm extends HttpServlet {

   /**
        *
        */
       private static final long serialVersionUID = 1130507799293250862L;

public void doGet(HttpServletRequest request, HttpServletResponse response)
      throws ServletException, IOException {

      // Create cookies for first and last names.
      Cookie firstName = new Cookie("first_name",
request.getParameter("first_name"));
```

```
      Cookie lastName = new Cookie("last_name",
request.getParameter("last_name"));
      Cookie campus = new Cookie("campus", request.getParameter("campus"));

      // Set expiry date after 24 Hrs for both the cookies.
      firstName.setMaxAge(60*60*24);
      lastName.setMaxAge(60*60*24);
      campus.setMaxAge(60*60*24);

      // Add both the cookies in the response header.
      response.addCookie( firstName );
      response.addCookie( lastName );
      response.addCookie( campus );

      // Set response content type
      response.setContentType("text/html");

      PrintWriter out = response.getWriter();
      String title = "Setting Cookies Example";
      String docType =
         "<!doctype html public \"-//w3c//dtd html 4.0 " +
"transitional//en\">\n";

      out.println(docType +
         "<html>\n" +
            "<head><title>" + title + "</title></head>\n" +
            "<body bgcolor = \"#f0f0f0\">\n" +
               "<h1 align = \"center\">" + title + "</h1>\n" +
               "<ul>\n" +
                  "  <li><b>First Name</b>: "
                  + request.getParameter("first_name") + "\n" +
                  "  <li><b>Last Name</b>: "
                  + request.getParameter("last_name") + "\n" +
                  "  <li><b>Campus</b>: "
                  + request.getParameter("campus") + "\n" +
               "</ul>\n" +
             "<a href=\"http://localhost:8080/Servlet-
Programming/ReadCookies\">"
             + "<button type=\"button\">Read Cookies</button></a>" +
            "</body></html>"
      );
   }
}
```

ReadCookies

```
// Import
required
java
libraries
```

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

// Extend HttpServlet class
public class ReadCookies extends HttpServlet {

   /**
        *
        */
       private static final long serialVersionUID = 1L;

public void doGet(HttpServletRequest request, HttpServletResponse response)
      throws ServletException, IOException {

      Cookie cookie = null;
      Cookie[] cookies = null;

      // Get an array of Cookies associated with this domain
      cookies = request.getCookies();

      // Set response content type
      response.setContentType("text/html");

      PrintWriter out = response.getWriter();
      String title = "Reading Cookies Example";
      String docType =
         "<!doctype html public \"-//w3c//dtd html 4.0 " +
         "transitional//en\">\n";

      out.println(docType +
         "<html>\n" +
         "<head><title>" + title + "</title></head>\n" +
         "<body bgcolor = \"#f0f0f0\">\n" );

      if( cookies != null ) {
         out.println("<h2> Found Cookies Name and Value</h2>");

         for (int i = 0; i < cookies.length; i++) {
            cookie = cookies[i];
            out.print("Name : " + cookie.getName( ) + ",  ");
```

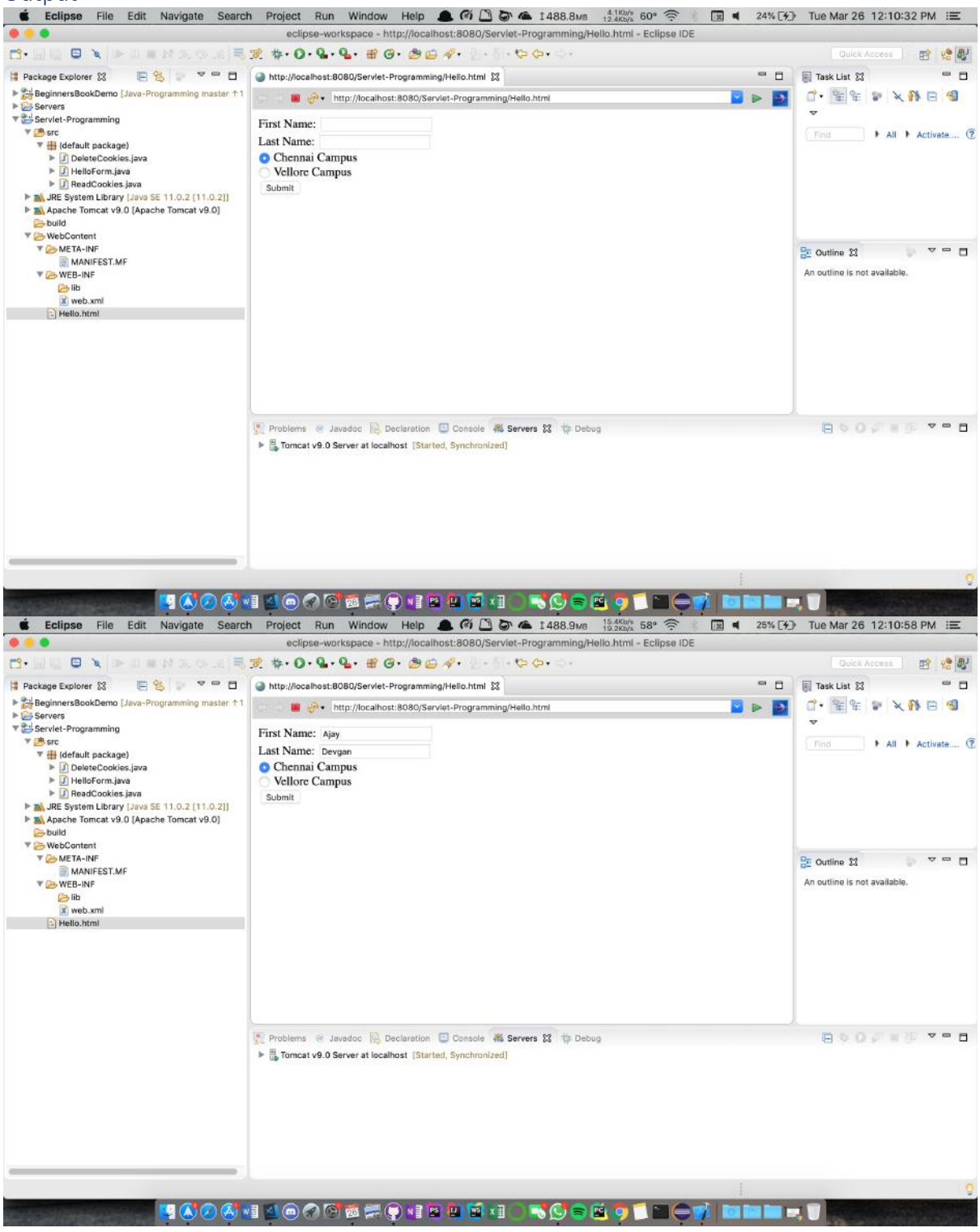
```
                  out.print("Value: " + cookie.getValue( ) + " <br/>");
               }
            } else {
               out.println("<h2>No cookies founds</h2>");
            }
            if( cookies != null ) {
                for (int i = cookies.length-1; i >= 0; i--) {
                   cookie = cookies[i];
                   if(cookie.getName().contains("first_name")) {
                      out.print("Welcome " + cookie.getValue() + " from ");
                   }
                   if(cookie.getName().contains("campus")) {
                      out.print(cookie.getValue() + " campus </br>");
                   }
                }
             }
            out.println("<a href=\"http://localhost:8080/Servlet-
Programming/DeleteCookies\">"
                  + "<button type=\"button\">Delete all cookies</button></a>");
            out.println("</body>");
            out.println("</html>");
         }
}
```

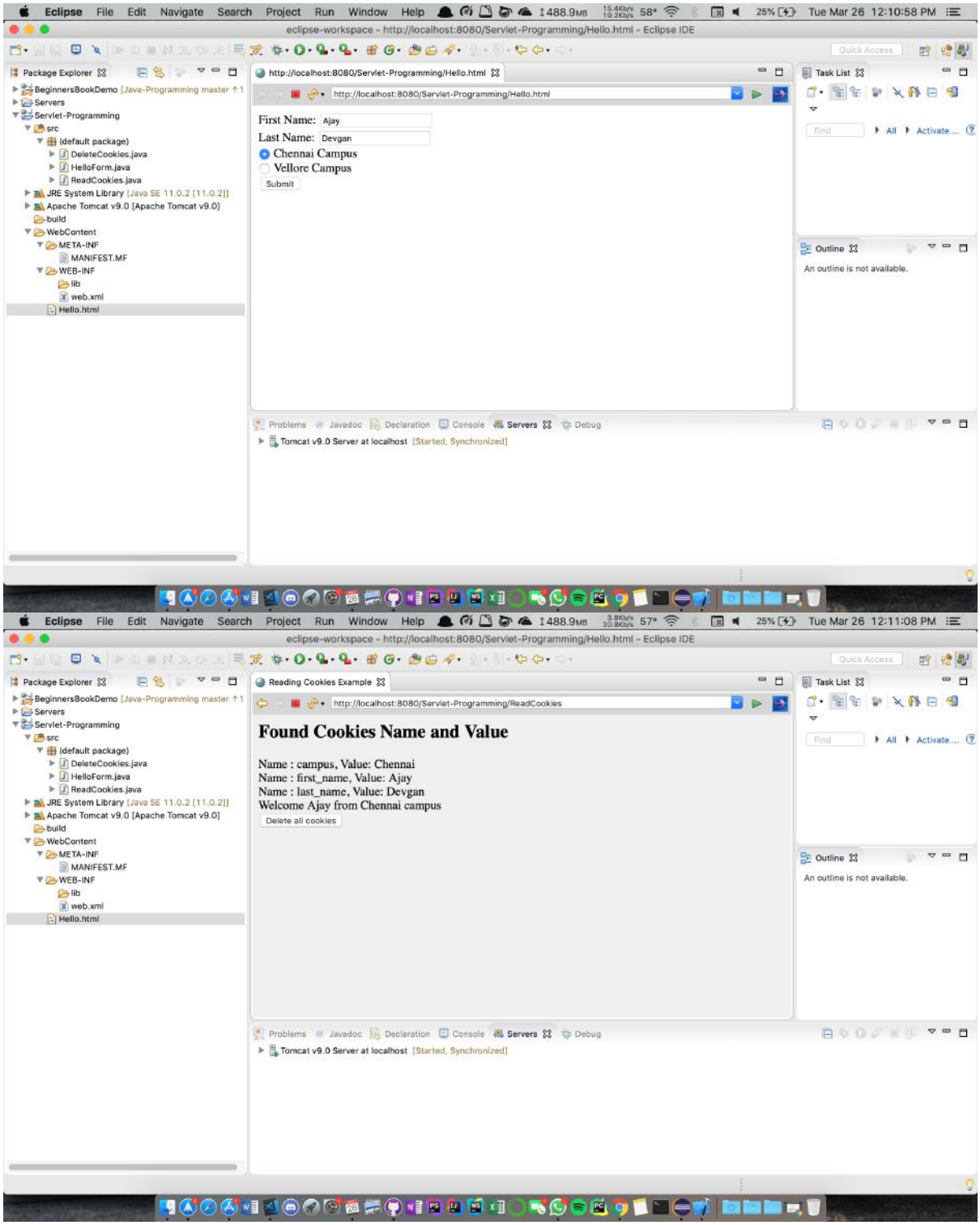
Delete Cookies

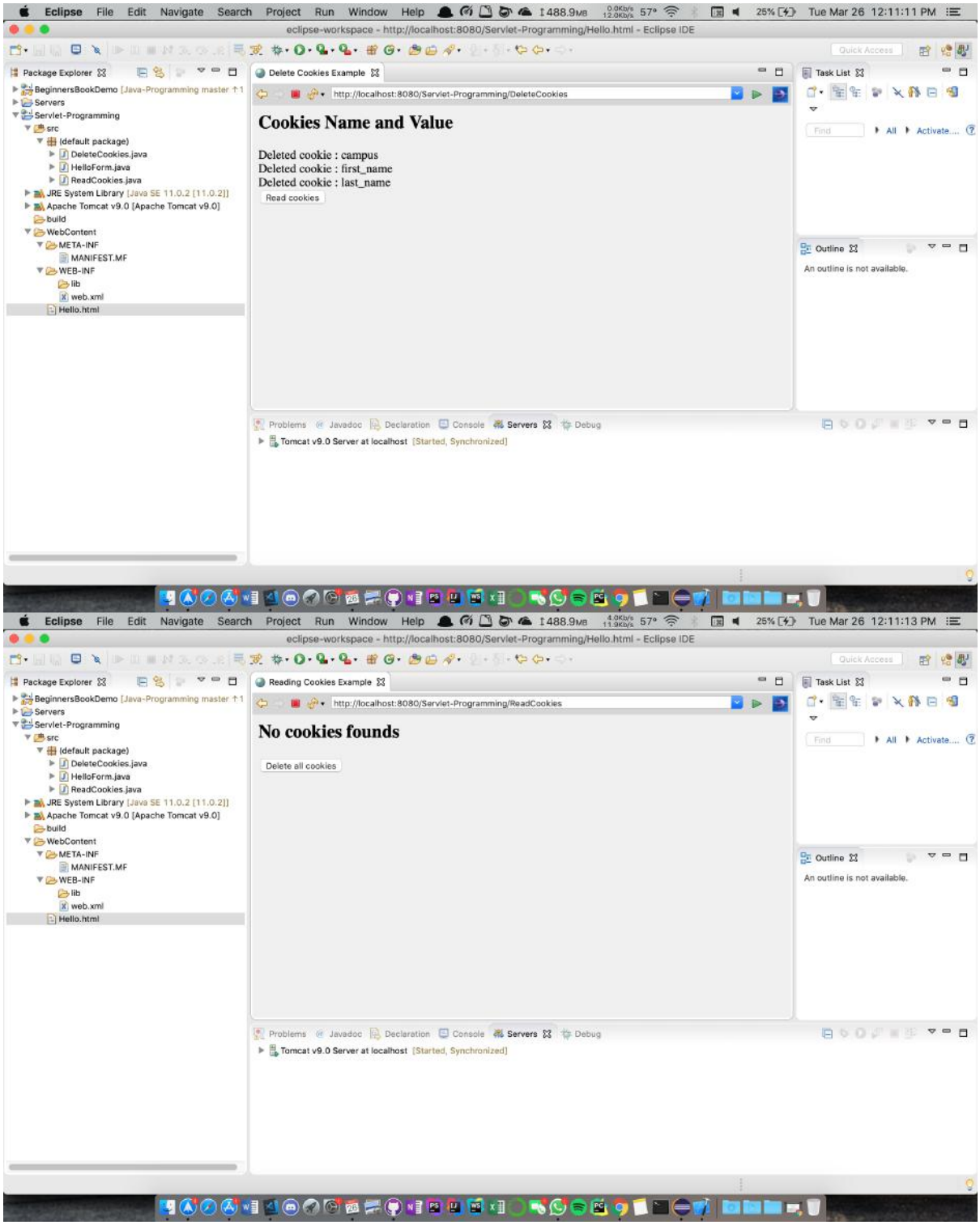
```
// Import
required
java
libraries
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

// Extend HttpServlet class
public class DeleteCookies extends HttpServlet {

   /**
        *
        */
       private static final long serialVersionUID = 1L;

public void doGet(HttpServletRequest request, HttpServletResponse response)
      throws ServletException, IOException {

      Cookie cookie = null;
```

```
        Cookie[] cookies = null;

        // Get an array of Cookies associated with this domain
        cookies = request.getCookies();

        // Set response content type
        response.setContentType("text/html");

        PrintWriter out = response.getWriter();
        String title = "Delete Cookies Example";
        String docType =
           "<!doctype html public \"-//w3c//dtd html 4.0 " +
"transitional//en\">\n";

        out.println(docType +
           "<html>\n" +
           "<head><title>" + title + "</title></head>\n" +
           "<body bgcolor = \"#f0f0f0\">\n" );

        if( cookies != null ) {
           out.println("<h2> Cookies Name and Value</h2>");

           for (int i = 0; i < cookies.length; i++) {
              cookie = cookies[i];
              cookie.setMaxAge(0);
              response.addCookie(cookie);
              out.print("Deleted cookie : " + cookie.getName( ) + "<br/>");
           }
        } else {
           out.println("<h2>No cookies founds</h2>");
        }
        out.println("<a href=\"http://localhost:8080/Servlet-
Programming/ReadCookies\">"
                 + "<button type=\"button\">Read cookies</button></a>");
        out.println("</body>");
        out.println("</html>");
     }
}
```

Output


eclipse-workspace - http://localhost:8080/Servlet-Programming/Hello.html - Eclipse IDE
Package Explorer
BeginnersBookDemo [Java-Programming master ↑1]
Servers
Servlet-Programming
src
(default package)
DeleteCookies.java
HelloForm.java
ReadCookies.java
JRE System Library [Java SE 11.0.2 [11.0.2]]
Apache Tomcat v9.0 [Apache Tomcat v9.0]
build
WebContent
META-INF
MANIFEST.MF
WEB-INF
lib
web.xml
Hello.html
http://localhost:8080/Servlet-Programming/Hello.html
First Name:
Last Name:
Chennai Campus
Vellore Campus
Submit
Task List
Outline
An outline is not available.
Problems
Javadoc
Declaration
Console
Servers
Debug
Tomcat v9.0 Server at localhost [Started, Synchronized]
First Name: Ajay
Last Name: Devgan


eclipse-workspace - http://localhost:8080/Servlet-Programming/Hello.html - Eclipse IDE
Package Explorer
BeginnersBookDemo [Java-Programming master ↑1]
Servers
Servlet-Programming
src
(default package)
DeleteCookies.java
HelloForm.java
ReadCookies.java
JRE System Library [Java SE 11.0.2 [11.0.2]]
Apache Tomcat v9.0 [Apache Tomcat v9.0]
build
WebContent
META-INF
MANIFEST.MF
WEB-INF
lib
web.xml
Hello.html
http://localhost:8080/Servlet-Programming/Hello.html
First Name: Ajay
Last Name: Devgan
Chennai Campus
Vellore Campus
Submit
Task List
Outline
An outline is not available.
Problems
Javadoc
Declaration
Console
Servers
Debug
Tomcat v9.0 Server at localhost [Started, Synchronized]
eclipse-workspace - http://localhost:8080/Servlet-Programming/Hello.html - Eclipse IDE
Reading Cookies Example
http://localhost:8080/Servlet-Programming/ReadCookies
Found Cookies Name and Value
Name : campus, Value: Chennai
Name : first_name, Value: Ajay
Name : last_name, Value: Devgan
Welcome Ajay from Chennai campus
Delete all cookies


eclipse-workspace - http://localhost:8080/Servlet-Programming/Hello.html - Eclipse IDE
Package Explorer
BeginnersBookDemo [Java-Programming master ↑1]
Servers
Servlet-Programming
src
(default package)
DeleteCookies.java
HelloForm.java
ReadCookies.java
JRE System Library [Java SE 11.0.2 [11.0.2]]
Apache Tomcat v9.0 [Apache Tomcat v9.0]
build
WebContent
META-INF
MANIFEST.MF
WEB-INF
lib
web.xml
Hello.html
Delete Cookies Example
http://localhost:8080/Servlet-Programming/DeleteCookies
Cookies Name and Value
Deleted cookie : campus
Deleted cookie : first_name
Deleted cookie : last_name
Read cookies
Task List
Outline
An outline is not available.
Problems
Javadoc
Declaration
Console
Servers
Debug
Tomcat v9.0 Server at localhost [Started, Synchronized]
eclipse-workspace - http://localhost:8080/Servlet-Programming/Hello.html - Eclipse IDE
Reading Cookies Example
http://localhost:8080/Servlet-Programming/ReadCookies
No cookies founds
Delete all cookies