



OBJECT ORIENTED PROGRAMMING

ISBN 978-93-6128-608-7



9 789361 286087

A.FAUZIA. M.C.A.,

OBJECT ORIENTED PROGRAMMING
WITH
SOLVED PROGRAMS

A. FAUZIA. M.C.A.,
DEPARTMENT OF COMPUTER SCIENCE
ENGINEERING

GENERAL GUIDELINES AND SAFETY INSTRUCTIONS

1. Sign in the log register as soon as you enter the lab and strictly observe your lab timings.
2. Strictly follow the written and verbal instructions given by the teacher / Lab Instructor. If you do not understand the instructions, the handouts and the procedures, ask the instructor or teacher.
3. Never work alone! You should be accompanied by your laboratory partner and / or the instructors / teaching assistants all the time.
4. It is mandatory to come to lab in a formal dress and wear your ID cards.
5. Do not wear loose-fitting clothing or jewels in the lab. Rings and necklaces are usual excellent conductors of electricity.
6. Mobile phones should be switched off in the lab. Keep bags in the bag rack.
7. Keep the labs clean at all times, no food and drinks allowed inside the lab.
8. Intentional misconduct will lead to expulsion from the lab.
9. Do not handle any equipment without reading the safety instructions. Read the handout and procedures in the Lab Manual before starting the experiments.
10. Do your wiring, setup, and a careful circuit checkout before applying power. Do not make circuit changes or perform any wiring when power is on.
11. Avoid contact with energized electrical circuits.
12. Do not insert connectors forcefully into the sockets.
13. NEVER try to experiment with the power from the wall plug.
14. Immediately report dangerous or exceptional conditions to the Lab instructor / teacher: Equipment that is not working as expected, wires or connectors are broken, the equipment that smells or “smokes”. If you are not sure what the problem is or what's going on, switch off the Emergency shutdown.
15. Never use damaged instruments, wires or connectors. Hand over these parts to the Lab instructor/Teacher.
16. Be sure of location of fire extinguishers and first aid kits in the laboratory.
17. After completion of Experiment, return the bread board, trainer kits, wires, CRO probes and other components to lab staff. Do not take any item from the lab without permission.
18. Observation book and lab record should be carried to each lab. Readings of current lab experiment are to be entered in Observation book and previous lab experiment should be written in Lab record book. Both the books should be corrected by the faculty in each lab.
19. Special Precautions during soldering practice
 - a. Hold the soldering iron away from your body. Don't point the iron towards you.
 - b. Don't use a spread solder on the board as it may cause short circuit.
 - c. Do not overheat the components as excess heat may damage the components/board.
 - d. In case of burn or injury seek first aid available in the lab or at the college dispensary

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING
BASIC OBJECT ORIENTED PROGRAMMING

Mrs.A. Fauzia., M.C.A.,

Assistant Professor/ Computer Science and Engineering

Annai Mira College of Engineering and Technology Ranipet – 632 517

PREFACE

This book on “OBJECT ORIENTED PROGRAMMING PROGRAMS (CSE ,AI&DS , IT)” all department students can practices and the can improve them programming knowledge B.E[CSE],B.TECH[AI&DS], B.TECH[IT]. Degree course under Outcome Based Education Credit System with the new regulation 2021. This book covers bsic programs and connection with TCP,UDP, Cookies connectivity, Server- Client connection, CRUD, JDBC, ODBC, SMTP,SWING COMPONENTS, JQuery programming Language. We hope that this book will be useful to both teachers and students. Finally we would request the readers to kindly send their valuable comments and suggestions towards the improvement of the book and the same will be gratefully acknowledge. Any suggestion from the reader for the betterment of this book can be dropped into khizarfauzia12@gmail.com

Mrs.A.FAUZIA., MCA.,

ACKNOWLEDGEMENT

We are thankful to and fortunate enough to get constant encouragement, support and guideline from Chairman **Thiru.S.Ramadoss Ayya**, Secretary & Treasurer **Mr.G.Thamothiran** for his blessings to complete the book successfully. We would not forget to remember our Principal **Dr.T.K.Gopinathan.,Ph.D.**, for his constant assistance in preparing this book.

ANNAI MIRA COLLEGE OF ENGINEERING & TECHNOLOGY

S.NO	NAME OF THE PROGRAM	PAGE NO
1	Fibonacci Series in Java without using recursion	1
2	Prime Number Program in Java	2
3	How to Print ASCII Value in Java	3
4	Making Pattern Program	4
5	Using socket dateclient	5
6	Using socket dateserver	5
7	UDP Connection Datagram Socket udpsend	7
8	UDP Connection Datagram Socket Receiver udpreceive	8
9	Connection with using OOB Server	9
10	Connection with using OOB Client	10
11	Digital Networking and Storage (Cyclic Redundancy Check) Client	12
12	Digital Networking and Storage (Cyclic Redundancy Check) Server	13
13	Client Application	15
14	Server Application	17
15	AddServer Interface	19
16	AddServer Implimentation	19
17	AddServer	19

18	AddClient	20
19	Web Server	21
20	Web Client	22
21	Cookie Client	24
22	Cookie Server	25
23	Java Database Connectivity	27
24	Object Bank	30
25	Object Client	31
26	Object Server	33
27	Telnet Client	37
28	Telnet Sever	38
29	Simple Mail Transfer Protocol Server	40
30	Simple Mail Transfer Protocol Client	41
31	Standard Client	43
32	Standard Server	43
33	Standard implementation	44
34	Standard Interface	45
35	Implementation of Serialization	45
36	Package Java Database Connectivity	47
37	Using Menu's and Frame	51
38	The creation and display of the JSlider control is explained in Program.	53

39	Swing Slider	54
40	Swing –Standard Color	55
41	Creating Login page	57

Java Basic Programs

1. Fibonacci Series in Java without using recursion

```
class FibonacciExample
{
    public static void main(String args[])
    {
        int n1=0,n2=1,n3,i,count=10;
        System.out.print(n1+" "+n2);
        for(i=2;i<count;++i)
        {
            n3=n1+n2;
            System.out.print(" "+n3);
            n1=n2;
            n2=n3;
        }
    }
}
```

Output: 0 1 1 2 3 5 8 13 21 34

2. Prime Number Program in Java

```
public class PrimeExample
{
    public static void main(String args[])
    {
        int i ,m=0,flag=0;
        int n=3;
        m=n/2;
        if(n==0||n==1)
        {
            System.out.println(n+" is not prime number");
        }else{
            for(i=2;i<=m;i++){
                if(n%i==0){
                    System.out.println(n+" is not prime number");
                    flag=1;
                    break;
                }
            }
            if(flag==0) { System.out.println(n+" is prime number");
            }
        }
    }
}
```

Output : 3 is prime Number

3. How to Print ASCII Value in Java

```
public class PrintAsciiValueExample1
{
public static void main(String[] args)
{
char ch1 = 'a';
char ch2 = 'b';
int asciivalue1 = ch1;
int asciivalue2 = ch2;
System.out.println("The ASCII value of " + ch1 + " is: " + asciivalue1);
System.out.println("The ASCII value of " + ch2 + " is: " + asciivalue2);
}
}
```

Output: The ASCII value of a is: 97
The ASCII value of b is: 98

4. Making Pattern Program

```
public class pattern{
public static void main(String []args){
int lines=8;
int i,j;
for(i=1;i<lines;i++){
    for(j=1;j<=lines/2;j++){
        if(i==j){
            System.out.print(j);
        }else if(i>4 && j==lines-i){
            System.out.print(j); }
        else{
            System.out.print(" "); }
            j=j-2;
        while(j>0){
            if(i==j){
                System.out.print(j); }
            else if(i>4 && j==lines-i){
                System.out.print(j);
            }
            else{
                System.out.print(" "); }
            j--;
        }
        System.out.println("");
    }
}
}
```

Output:

```
1 1
2 2
3 3
4
3 3
2 2
1 1
```

5. Using socket dateclient

```
import java.net.*;
import java.io.*;
class dateclient
{
public static void main(String args[])
{
    Socket soc;
    DataInputStream dis;
    String sdate;
    PrintStream ps;
    try
    {
        InetAddress ia=InetAddress.getLocalHost();
        soc=new Socket(ia,8020);
        dis=new DataInputStream(soc.getInputStream());
        sdate=dis.readLine();
        System.out.println("The date in the server is : "+sdate);
        ps=new PrintStream(soc.getOutputStream());
        ps.println(ia);
    }
    catch(IOException e)
    {
        System.out.println("The exception is : "+e);
    }
}
}
```

6. Using Socket dateserver

```
import java.net.*;
import java.io.*;
import java.util.*;
class dateserver
{
public static void main(String args[])
{
    ServerSocket ss;
    Socket s;
    PrintStream ps;
    DataInputStream dis;
    String inet;
```

```

try
{
    ss=new ServerSocket(8020);
    while(true)
    {
        s=ss.accept();
        ps=new PrintStream(s.getOutputStream());
        Date d=new Date();
        ps.println(d);
        dis=new DataInputStream(s.getInputStream());
        inet=dis.readLine();
        System.out.println("The Client System Address is : "+inet);
        ps.close();
    }
}
catch(IOException e)
{
    System.out.println("The exception is : "+e);
}
} }

```

Output:

```

D:\JDK\BIN>javac dateserver.java
Note: dateserver.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
D:\JDK\BIN>java dateserver
D:\JDK\BIN>The Client System Address is : sys8/197.168.1.76
D:\JDK\BIN>javac dateclient.java
D:\JDK\BIN>Note: dateclient.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
D:\JDK\BIN>java dateclient
The date in the server is : THU Oct 8 13:42:39 GMT+05:30 2009

```

7. UDP Connection Datagram Socket udpsend

```
import java.net.*;
import java.io.*;
class udpsend
{
public static void main(String args[])
{
    String hostname;
    if (args.length>0)
    {
        hostname=args[0];
    }
    else
    {
        hostname="LocalHost";
    }
    int port=8050;
    String thisline;
    byte[] buffer=new byte[65535];
    try
    {
        DatagramSocket ds=new DatagramSocket();
        DataInputStream dis=new DataInputStream(System.in);
        while (true)
        {
            System.out.println("Enter the message to be sent \n");
            thisline=dis.readLine();
            if(thisline.equals("."))
                break;
            byte[] data=new byte[thisline.length()];
            thisline.getBytes(0,thisline.length(),data,0);
            DatagramPacket dp=new DatagramPacket(data,data.length,
            InetAddress.getByName(hostname),port);
            ds.send(dp);
            DatagramPacket dpl=new DatagramPacket(buffer,buffer.length);
            ds.receive(dpl);
            String s=new String(dpl.getData(),0,0,dpl.getLength());
```



```

        System.out.println("The Echo Message Received is \n" +s);
    }
}
catch(SocketException e1)
{
    System.out.println("Socket Exception "+e1);
}

catch(IOException e2)
{
    System.out.println("IO Exception Occured "+e2);
}
}
}

```

8. UDP Connection Datagram Socket Receiver udpreceive

```

import java.net.*;
import java.io.*;
class udpreceive
{
public static void main(String args[])
{
byte[] buffer=new byte[65535];
byte[] buffer1=new byte[65535];
try
{
String thisline;
String hostname;
hostname="LocalHost";
int port=8050;
DatagramSocket ds=new DatagramSocket(port);
while(true)
{
DatagramPacket dp=new DatagramPacket(buffer,buffer.length);
ds.receive(dp);
String s=new String(dp.getData(),0,0,dp.getLength());
InetAddress ia=dp.getAddress();
System.out.println("The message received from the host " +ia+ " is\n" +s);
int p=dp.getPort();
System.out.println("Received from Port : "+p);
buffer1=new byte[s.length()];
s.getBytes(0,s.length(),buffer1,0);
DatagramPacket dd=new DatagramPacket(buffer1,buffer1.length, ia,p);
ds.send(dd);
}
}
}

```

```

}
    }
    catch(IOException e1)
    {
        System.out.println("IO Exception Occured " +e1);
    }
    catch(Exception e2)
    {
        System.out.println("Socket Exception occured " +e2);
    } }
}

```

Output:

udpsend.java

D:\JDK\BIN>javac udpclient.java

D:\JDK\BIN>Note: udpclient.java uses or overrides a deprecated API.

Note: Recompile with -Xlint:deprecation for details.

D:\JDK\BIN>java udpclient

Enter the message to be sent

Hello

The Echo Message Received is

Hello

Enter the message to be sent

udpreceive.java

D:\JDK\BIN>javac udpserver.java

D:\JDK\BIN>Note: udpserver.java uses or overrides a deprecated API.

Note: Recompile with -Xlint:deprecation for details.

D:\JDK\BIN>java udpserver

The message received from the host /127.0.0.1 is

Hello

Received from Port : 2430

9. Connection with using OOB Server and Client

oobserver.java

```
import java.net.*;
```

```
import java.io.*;
```

```
class oobserver
```

```
{    public static void main(String args[])
```

```
{
```

```
    byte []buffer=new byte[65535];
```

```
    byte []buffer1=new byte[65535];
```

```

try
    {
        String thisline;
        String hostname;
        hostname="LocalHost";
        int port=8020;
        DatagramSocket ds=new DatagramSocket(port);
        DataInputStream dis=new DataInputStream(System.in);
        while(true)
        {
            DatagramPacket dp=new DatagramPacket(buffer,buffer.length);
            ds.receive(dp);
            String s=new String(dp.getData(),0,4,dp.getLength()-4);
            InetAddress ia=dp.getAddress();
            System.out.println("The message received from the host "+ia+"is\n"+s);
            int p=dp.getPort();
            System.out.println("Received from port : "+p);
            thisline=dis.readLine();
            if(thisline.equals("."))
                break;
            byte[] data=new byte[thisline.length()];
            thisline.getBytes(0,thisline.length(),data,0);
            DatagramPacket dd=new DatagramPacket(data,data.length,ia,p);
            ds.send(dd);
        }
        ds.close();
    }
catch(IOException e1)
{
    System.out.println("IO Exception Occured"+e1);
}
catch(Exception e2)
{
    System.out.println("Socket Exception Occured"+e2);
} } }

```

10. Connection with using OOB Client

```

oobclient.java
import java.net.*;
import java.io.*;
class oobclient {

```

```

public static void main(String args[])
{
String hostname;
String s;
try
{
if((args.length)>0)

        hostname=args[0];
else
        hostname="LocalHost";
        int port=8020;
        byte[] buffer=new byte[65535];
        byte oob=(byte) 0xff;
        String thisline;
DatagramSocket ds1=new DatagramSocket();
DataInputStream dis=new DataInputStream(System.in);
while(true)
{
        thisline=dis.readLine();
        if(thisline.equals("."))
        break;
        byte[] data=new byte[thisline.length()+4];
        for(int i=0;i<4;i++){
                data[i]=oob;    }
        thisline.getBytes(0,thisline.length(),data,4);
DatagramPacket dp1=new DatagramPacket(data,data.length,InetAddress.getByName(hostname),port);
ds1.send(dp1);
DatagramPacket dp=new DatagramPacket(buffer,buffer.length);
ds1.receive(dp);
s=new String(dp.getData(),0,0,dp.getLength());
System.out.println(s);
} ds1.close();
} catch(SocketException e1){
        System.out.println("Socket Exception"+e1);
} catch(IOException e2){
        System.out.println("IO Exception Occured"+e2);
}/*catch(UnknownHostException e){
        System.out.println(e);
}*/

```

```
}  
}
```

Output

```
D:\JDK\BIN>javac OobServer.java
```

```
D:\JDK\BIN>java OobServer
```

```
Hi
```

```
1075
```

```
good
```

```
morning
```

```
1075
```

```
D:\JDK\BIN>javac OobClient.java
```

```
D:\JDK\BIN>java OobClient
```

```
Hi
```

```
good
```

```
morning
```

11. Digital Networking and Storage (Cyclic Redundancy Check) Client

```
import java.io.*;  
import java.util.zip.*;  
import java.net.*;  
class crcclient  
{  
public static void main(String args[])  
{  
    try  
    {  
        InetAddress ia=InetAddress.getLocalHost();  
        CRC32 cc=new CRC32();  
        System.out.println("Please enter a message");  
        BufferedReader br=new BufferedReader(new InputStreamReader (System.in));  
        String s=br.readLine();  
        PrintStream ps;  
        String m;  
        Socket ss=new Socket(ia,8000);  
        cc.update(s.getBytes());  
        long val=cc.getValue();  
        ps=new PrintStream(ss.getOutputStream());  
        m=s+"//"+val;  
        System.out.println("The message and the crc value is ");
```

```

System.out.println(m);
ps.println(m);
DataInputStream dis=new DataInputStream(ss.getInputStream());
String str=dis.readLine();
System.out.println("The Acknowledgement received from server is
\n");
System.out.println(str);
    }
    catch(Exception e)
    {
        System.out.println(e);
    }
}
}
}

```

12. Digital Networking and Storage (Cyclic Redundancy Check)

crcserver.java

```

import java.util.zip.*;
import java.util.*;
import java.io.*;
import java.net.*;
class crcserver
{
public static void main(String args[])
{
    try
    {
        CRC32 c=new CRC32();
        ServerSocket ss=new ServerSocket(8000);
        Socket s;
        String str=" ";
        String sis[]=new String[500];
        while(true)
        {
            s=ss.accept();
            DataInputStream dis=new DataInputStream(s.getInputStream());
            str=dis.readLine();
            System.out.println("The message with the CRC value received is \n");
            System.out.println(str);
            StringTokenizer st=new StringTokenizer(str,"/");
            int n=st.countTokens();
            System.out.println("Number of token" +n);
            System.out.println("The Tokens received are ");
            for (int i=0;i<n;i++)
            { sis[i]=st.nextToken();

```

```

        System.out.println(sis[i]);
    }
    long val1=Long.parseLong(sis[n-1]);
    String s1=Long.toString(val1);

    System.out.println("The calculate CRC value is ");
    System.out.println(val1);
    c.update(sis[0].getBytes());
    long val=c.getValue();
    String s2=Long.toString(val);
    if(s1.equals(s2))
    {
        String s3="Good Message";
        PrintStream ps=new PrintStream(s.getOutputStream());
        ps.println(s3);
    }
    System.out.println("The message from client is accepted");
    } }
    catch(Exception e)
    {
        System.out.println(e);
    }
}
}

```

Output:

//crclient.java

D:\JDK\BIN>javac crclient.java

D:\JDK\BIN>Note: crclient.java uses or overrides a deprecated API.

Note: Recompile with -Xlint:deprecation for details.

D:\JDK\BIN>java crclient

Please enter a message

D:\JDK\BIN>Happy Birthday

The message and the crc value is

Happy Birthday//4188431648

The Acknowledgement received from server is

Good Message

//crcserver.java

D:\JDK\BIN>javac crcserver.java

Note: crcserver.java uses or overrides a deprecated API.

Note: Recompile with -Xlint:deprecation for details.

D:\JDK\BIN>java crcserver

D:\JDK\BIN>The message with the CRC value received is

Happy Birthday//4188431648
Number of token2
The Tokens received are
Happy Birthday
4188431648
The calculate CRC value is
4188431648
The message from client is accepted.

13. Client Application

```
import java.awt.*;
import java.awt.event.*;
import java.net.*;
import java.io.*;
public class client extends Frame implements ActionListener,Runnable
{
    Socket s;
    BufferedReader br;
    BufferedWriter bw;
    TextField text;
    Button button1,button2;
    List list;
    public static void main(String args[])
    {
        new client("client application");
    }
    public void run()
    {
        try{
            s.setSoTimeout(1);
        }
        catch(Exception e) { }
    while(true)
    {
        try
        {
            list.addItem(br.readLine());
        }
        catch(Exception h) { }
    }
    }
    public client (String m)
    {
        super(m);
```



```

setSize(200,300);
setLocation(300,0);
this.setLayout(new BorderLayout());button1=new Button("send");
button2=new Button("exit");
button1.addActionListener(this);
button2.addActionListener(this);
list=new List();
text=new TextField();
add(list,"Center");
add(button1,"West");
add(button2,"East");
add(text,"South");
setVisible(true);
try
{
    s=new Socket("127.0.0.1",100);
    br=new BufferedReader(new InputStreamReader
(s.getInputStream()));
    bw=new BufferedWriter(new OutputStreamWriter
(s.getOutputStream()));
    Thread th;
    th=new Thread(this);
    th.start();
}catch(Exception e) { }
    }    public void actionPerformed(ActionEvent e)
        {
            if(e.getSource().equals(button2))
                System.exit(0);
            else
            {
                try
                {
                    bw.write(text.getText());
                    bw.newLine();
                    bw.flush();
                }catch(Exception m)
                { }
            }
        }
    }
}

```

14. Server Application

```
import java.awt.*;
import java.awt.event.*;
import java.net.*;
import java.io.*;
public class serverapp extends Frame implements ActionListener,Runnable

{

ServerSocket s;
Socket s1;
BufferedReader br;
BufferedWriter bw;
TextField text;
Button button1,button2;
List list;
public static void main(String args[])
{
    new serverapp("server application");
} public void run()
{
    try
    {
        s.setSoTimeout(1);
    }
    catch(Exception e)
    { } while(true)
    {
        try
        {
            list.addItem(br.readLine());
        } catch(Exception h) { }
    }
} public serverapp(String m)

{
super(m);
setSize(200,300);
setLocation(300,0);
this.setLayout(new BorderLayout());button1=new Button("send");button2=new Button("exit");
```

```

button1.addActionListener(this);
button2.addActionListener(this);
list=new List();
text=new TextField();
add(list,"Center");
add(button1,"West");
add(button2,"East");
add(text,"South");
setVisible(true);
try
{
s=new ServerSocket(100);
s1=s.accept();;
br=new BufferedReader(new InputStreamReader(s1.getInputStream()));
bw=new BufferedWriter(new OutputStreamWriter(s1.getOutputStream()));
Thread th;
th=new Thread(this);
th.start();
}
catch(Exception e)
{ }
}
public void actionPerformed(ActionEvent e)
{
if(e.getSource().equals(button2))
    System.exit(0);
else
{
    try
    {
        bw.write(text.getText());
        bw.newLine();
        bw.flush();
    }
    catch(Exception m)
    { }
} }
}

```

Output:

D:\JDK\BIN>javac client.java

Note: crserver.java uses or overrides a deprecated API.

Note: Recompile with -Xlint:deprecation for details.

D:\JDK\BIN>javac serverapp.java

Note: crserver.java uses or overrides a deprecated API.

Note: Recompile with -Xlint:deprecation for details.



15. AddServer Interface

```
import java.rmi.*;
public interface AddServerIntf extends Remote
{
    double add(double d1,double d2)throws RemoteException ;
}
```

16. AddServer Implimentation

```
import java.rmi.*;
import java.rmi.server.*;
public class AddServerImpl extends UnicastRemoteObject implements AddServerIntf
{
    public AddServerImpl() throws RemoteException
    }
    public double add(double d1,double d2)throws RemoteException
    {
        return (d1+d2);
    }
}
```

17. AddServer

```
import java.net.*;
import java.rmi.*;
public class AddServer
{
    public static void main(String args[]) {
```

```

try
{
    AddServerImpl addserverImpl=new AddServerImpl();
    Naming.rebind("AddServer",addserverImpl);
}
    catch(Exception e)
        {
            System.out.println("Exception"+e);
        }
}
}

```

18. Add Client

```

import java.net.*;
import java.rmi.*;
public class Addclient
{
    public static void main(String args[])
    {
        try
        {
            String addserverUrl="rmi://" +args[0]+"/AddServer";
            AddServerIntf intf=(AddServerIntf)Naming.lookup(addserverUrl);
            System.out.println("The first number is"+args[1]);
            double d1=Double.valueOf(args[1]).doubleValue();
            System.out.println("The second number is"+args[2]);
            double d2=Double.valueOf(args[2]).doubleValue();
            System.out.println("the sum is"+intf.add(d1,d2));
        }
        catch(Exception e)
        {
            System.out.println("Exception "+e);
        }
    }
}
}

```

Output:

//Addserver.java

D:\JDK\BIN>javac AddServerIntf.java

D:\JDK\BIN>javac AddServerImpl.java

D:\JDK\BIN>javac Addclient.java

D:\JDK\BIN>javac AddServer.java

D:\JDK\BIN>rmic AddServerImpl

D:\JDK\BIN>start rmiregistry

D:\JDK\BIN>java AddServer

Addclient.java

D:\JDK\BIN>java Addclient localhost 8 9

The first number is: 8

The second number is: 9

The sum is: 17.0

19. Web Server

```
import java.io.*;
import java.net.*;
class webservice
{
public static void main(String args[])
{
    ServerSocket ss=null;
    Socket s=null;
    int connects=0;
    try
    {
        ss=new ServerSocket(80,5);
        while (connects < 5)
        {
            s=ss.accept();
            ServiceClient(s);
            connects++;
        }
        ss.close();
    }
    catch (Exception e)
    {
        System.out.println(e);
    }
}
public static void ServiceClient(Socket client) throws IOException
{
    DataInputStream dis = null;
    DataOutputStream dos = null;
    try
    {
        dis = new DataInputStream(client.getInputStream());
        dos = new DataOutputStream(client.getOutputStream());
```

```

StringBuffer buf = new StringBuffer("<html><body><p>Welcome to Simple Web Page
</body></html>");
String str;
while ((str=dis.readLine())!=null)
{
    if(str.equals(""))
    {
        dos.writeBytes(buf.toString());
        break;
    }
}
}
finally
{
    System.out.println("Cleaning up Connection : " +client);
    dos.close();
    dis.close();
    client.close();
}
}
}

```

20. Web Client

```

import java.io.*;
import java.net.*;
class webclient
{
public static void main (String args[])
{
    try
    {
        Socket s1 = new Socket ("localhost",80);
        System.out.println("Client1 : " +s1);
        getPage(s1);
    }
    catch (UnknownHostException e)
    {
        System.out.println(e);
    }
    catch (IOException e)

```

```

        {
            System.out.println(e);
        }
    }public static void getPage(Socket s)
    {
        try
        {
            DataOutputStream dos = new DataOutputStream(s.getOutputStream());
            DataInputStream dis = new DataInputStream(s.getInputStream());
            dos.writeBytes("GET/HTTP/1.0\r\n\r\n");
            String responseLine;
            while((responseLine=dis.readLine())!=null)
            {
                System.out.println(responseLine);
                if(responseLine.indexOf("<html>")!=-1)
                    break;
            }
            dos.close();
            dis.close();
            s.close();
        }
        catch(IOException e)
        {
            System.out.println(e);
        }
    }
}

```

Output:

//webserver.java

D:\JDK\BIN>javac webserver.java

D:\JDK\BIN>Note: webserver.java uses or overrides a deprecated API.

Note: Recompile with -Xlint:deprecation for details.

D:\JDK\BIN>java webserver

D:\JDK\BIN>

Cleaning up Connection : Socket[addr=/127.0.0.1,port=4490,localport=80]

Cleaning up Connection : Socket[addr=/127.0.0.1,port=4491,localport=80]

//webclient.java

D:\JDK\BIN>javac webclient.java

D:\JDK\BIN>Note: webclient.java uses or overrides a deprecated API.

Note: Recompile with -Xlint:deprecation for details.

D:\JDK\BIN>java webclient

D:\JDK\BIN>Client1 : Socket[addr=localhost/127.0.0.1,port=80,localport=4490]

<html><body><p>Welcome to Simple Web Page </body></html>

D:\JDK\BIN>java webclient

D:\JDK\BIN>Client1 : Socket[addr=localhost/127.0.0.1,port=80,localport=4491]

<html><body><p>Welcome to Simple Web Page </body></html>

21. Cookie Client

```
import java.io.*;
import java.net.*;
import java.util.*;
class cookieclient
{
    public static void main(String args[]) throws Exception
    {
        Socket soc=null;
        DataInputStream dis=null;
        DataOutputStream dos=null;
        try
        {
            System.out.println("Connecting to Server \n");
            soc=new Socket("LocalHost",8255);
            dis=new DataInputStream(soc.getInputStream());
            dos=new DataOutputStream(soc.getOutputStream());
            String Request="GET:home.html";
            System.out.println("Request sent:"+Request+"\n");
            dos.writeBytes(Request+"\n");
            String Resp=dis.readLine();
            System.out.println("Response:"+Resp+"\n");
            Thread.sleep(1000);
            Request="Cookie"+Resp.substring(10);
            System.out.println("Request send:"+Request+"\n");
            dos.writeBytes(Request+"\n");
            Resp=dis.readLine();
        }
    }
}
```

```

        System.out.println("Closing connections"); soc.close();    }
        catch(IOException e){System.out.println(e);    }
    }
}

```

22. Cookie Server

```

import java.io.*;
import java.net.*;
import java.util.*;
class cookieserver
{
    public static void main(String args[])
    {
        ServerSocket ss=null;
        Socket s=null;
        DataInputStream dis=null;
        DataOutputStream dos=null;
        try
        {
            System.out.println("Server serialization");
            ss=new ServerSocket(8255);
            System.out.println("Waiting for request..");
            s=ss.accept();
            dis=new DataInputStream(s.getInputStream());
            dos=new DataOutputStream(s.getOutputStream());
            String Name="Test Cookie";
            String Value="ABCDEF";
            Date Expires=new Date();
            String Domain="localhost";
            String Path="/";
            String IsSecure="TRUE";
            String Cookie="Set-Cookie:Name="+Name+";
VALUE="+Value+";EXPIRES="+Expires.toString()+";
DOMAIN="+Domain+";PATH="+Path+";
ISSECURE="+IsSecure;
String Request=dis.readLine();
System.out.println("Request received:"+Request);
if(Request.startsWith("GET"))
{
    System.out.println("Send:"+Request.substring(4)+"\n");
    System.out.println("Send:"+Cookie+"\n");

```

```

        dos.writeBytes(Cookie+"\n");    }
Request=dis.readLine();
System.out.println("Request received:"+Request+"\n");
if(Request.startsWith("Cookie"))
{
Cookie="Set-
Cookie:Name="+Name+";VALUE="+Value+";EXPIRES="+Expires.toString()+";DOMAIN="+Domain+";PATH="+Path+";ISSECURE="+IsSecure;
System.out.println("Send:"+Cookie+"\n");
dos.writeBytes(Cookie+"\n");
}
System.out.println("Closing connection");
}
catch(IOException e){System.out.println(e);}
} }

```

Output:

//cookieserver.java

D:\JDK\BIN>javac cookieserver.java

D:\JDK\BIN>java cookieserver

Server serialization

Waiting for request..

Request received:GET:home.html

Send:home.html

Send:Set-Cookie:Name=Test Cookie;VALUE=ABCDEF;EXPIRES=Fri Oct 26 14:00:10 IST 2007;DOMAIN=localhost;PATH=/;ISSECURE=TRUE

Request received:Cookie:Name=Test Cookie;VALUE=ABCDEF;EXPIRES=Fri Oct 26 14:00:10 IST 2007;DOMAIN=localhost;PATH=/;ISSECURE=TRUE

Send:Set-Cookie:Name=Test Cookie;VALUE=ABCDEF;EXPIRES=Fri Oct 26 14:00:10 IST 2007;DOMAIN=localhost;PATH=/;ISSECURE=TRUE

Closing connection

//cookieclient.java

D:\JDK\BIN>javac cookieclient.java

D:\JDK\BIN>java cookieclient

Connecting to Server

Request sent:GET:home.html

Response:Set-Cookie:Name=Test Cookie;VALUE=ABCDEF;EXPIRES=Fri Oct 26 14:00:10 IST 2007;DOMAIN=localhost;PATH=/;ISSECURE=TRUE

Request send:Cookie:Name=Test Cookie;VALUE=ABCDEF;EXPIRES=Fri Oct 26 14:00:10 IST 2007;DOMAIN=localhost;PATH=/;ISSECURE=TRUE

Closing connections

23. Java Database Connectivity

```
import java.io.*;
import java.sql.*;
class Jdbc
{
    public static void main(String args[])
    {
        ResultSet rs=null;
        String stud_no=" ";
        String choice=" ";
        DataInputStream dis=new DataInputStream(System.in);
        try
        {
            display_line();
            System.out.println("\n");
            System.out.println("Enter ur choice:");
            System.out.println("Display single record(1):");
            System.out.println("Display all records(2):");
            choice=dis.readLine();
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
            Connection con=DriverManager.getConnection("jdbc:odbc:stud");
            Statement stmt=con.createStatement();
            if(choice.equals("1"))
            {
                System.out.println("\n");
                System.out.println(" Entered choice is one");
                System.out.println("Enter the student register no:");
                stud_no=dis.readLine();
                int j=Integer.parseInt(stud_no);
                rs=stmt.executeQuery("select * from student where regno="+j+"");
            }
            if(choice.equals("2"))
            {
                System.out.println("\n");
                System.out.println("\t Entered choice is two");
                rs=stmt.executeQuery("select * From student");
            }while(rs.next())
            {
                System.out.println("\n");
                System.out.println(rs.getString(1));
            }
        }
    }
}
```

```

        System.out.println(rs.getString(2));
    }
    display_line();
}
catch(Exception e)
{
    System.out.println(e);
}
}
public static void display_line()
{
    char ch='*';
    int i=0;
    for(i=0;i<75;i++)
    {
        System.out.print(ch);
    }
}
}

```

Output:

//Student.mdb

Student	
RegNo	Name
1	Anitha
2	Banu
3	Chitra
4	Dafne
5	Geetha

//Jdbc.java

D:\JDK\BIN >javac Jdbc.java

D:\JDK\BIN >java Jdbc

Enter ur choice:

Display single record(1):

Display all records(2):

1

Entered choice is one

Enter the student register no:

1

1

Anitha

D:\JDK\BIN >java Jdbc

Enter ur choice:

Display single record(1):

Display all records(2):

2

Entered choice is two

1

Anitha

2

Banu

3

Chitra

4

Dafne

5

Geetha

24. Object Bank

```
import java.io.*;
import java.sql.*;
public class ObjectBank implements Serializable
{
    String Accno,Name,Gender,Balance,Date,Address;
    public void bank()
    {
        Accno=null;
        Name=null;
        Gender=null;
        Balance=null;
        Date=null;
        Address=null;
    }
    public void setAccno(String accno)
    {
        Accno=accno;
    }
    public String getAccno()
    {
        return Accno;
    }
    public void setName(String n)
    {
        Name=n;
    }
    public String getName()
    {
        return Name;
    }
    public void setGender(String g)
    {
        Gender=g;
    }
    public String getGender()
    {
        return Gender;
    }
    public void setAddress(String ad)
    {
        Address=ad;
    }
}
```

```

    public String getAddress()
    {
        return Address;
    }
    public void setDate(String d)
    {
        Date=d;
    }
public String getDate()
    {
        return Date;
    }
    public void setBalance(String b)
    {
        Balance=b;
    }
    public String getBalance()
    {
        return Balance;
    }
}

```

25. Object Client

```

import java.io.*;
import java.net.*;
import java.sql.*;
import java.util.*;
class ObjectClient
{
    public static void main(String args[])
    {
        Socket soc=null;
        PrintWriter pw=null;
        ObjectInputStream in=null;
        DataOutputStream dos=null;
        BufferedReader br=null;
        ObjectBank b=null;
        try

```



```

{
    soc=new Socket("localhost",8080);
    pw=new PrintWriter(new BufferedWriter(new
    OutputStreamWriter(soc.getOutputStream())),true);
    dos=new DataOutputStream(soc.getOutputStream());
    br=new BufferedReader(new InputStreamReader(System.in));
    System.out.println("Enter the choice : (all/recordno)");
    String input=br.readLine();
    dos.writeBytes(input + "\n");
    in=new ObjectInputStream(soc.getInputStream());
    if(input.startsWith("al"))
    {
        Vector v=null;
        v=(Vector)in.readObject();
        for(int i=0;i<v.size();i++)
        {
            b=(ObjectBank)v.elementAt(i);
            System.out.println("\t ***CUSTOMER DETAILS***");
            System.out.println(b.getAccno());
            System.out.println(b.getName());
            System.out.println(b.getGender());
            System.out.println(b.getAddress());
            System.out.println(b.getDate());
            System.out.println(b.getBalance());
        }
    }
    else
    {
        b=(ObjectBank)in.readObject();
        System.out.println("\t ***CUSTOMER DETAILS***");
        System.out.println(b.getName());
        System.out.println(b.getGender());
        System.out.println(b.getAddress());
        System.out.println(b.getDate());
        System.out.println(b.getBalance());
    }
}
} catch(Exception e)
    { e.printStackTrace(); }
}

```

26. Object Server

```
import java.io.*;
import java.net.*;
import java.sql.*;
import java.util.*;
class ObjectServer
{
    public static void main(String args[]) throws Exception
    {
        DataInputStream dis=null;
        ObjectOutputStream out=null;
        ServerSocket ss=new ServerSocket(8080);
        Socket s=null;
        Connection con;
        Statement stmt;
        ResultSet rs;
        ObjectBank b=null;
        String url;
        Vector bank=new Vector();
        while(true)
        {
            try
            {
                s=ss.accept();
                dis=new DataInputStream(s.getInputStream());
                out=new ObjectOutputStream(s.getOutputStream());
                String in=dis.readLine();
                System.out.println("Input :"+in);
                Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
                url="jdbc:odbc:banking";
                con=DriverManager.getConnection(url);
                stmt=con.createStatement(ResultSet.TYPE_SCROLL
                _SENSITIVE,ResultSet.CONCUR_UPDATABLE);
                rs=stmt.executeQuery("select * from bank");
                if(in.equals("all"))
                {
                    rs.first();
                }
            }
        }
    }
}
```

```

do
{
    b=new ObjectBank();
    b.setAccno(rs.getString(1));
    b.setName(rs.getString(2));
    b.setGender(rs.getString(3));
    b.setAddress(rs.getString(4));
    b.setDate(rs.getString(5));
    b.setBalance(rs.getString(6));
    bank.addElement(b);
}
while(rs.next());
out.writeObject(bank);
}
else
{
    b=new ObjectBank();
    rs.first();
    do
    {
        String temp=rs.getString(1);
        if(in.equals(temp))
        {
            b.setName(rs.getString(2));
            b.setGender(rs.getString(3));
            b.setAddress(rs.getString(4));
            b.setDate(rs.getString(5));
            b.setBalance(rs.getString(6));
        }
    }
    while(rs.next());
    out.writeObject(b);
}
}catch(Exception e)
{
    e.getMessage();
}
}
}}

```

Output:

//Banking.mdb

Bank					
AccNo	Name	Gender	Address	Date	Balance
1	Anitha	F	Kodambakkam	10/9/2007	10000
2	Banu	F	Tnagar	10/9/2007	15000
3	Chitra	F	Kodambakkam	10/9/2007	30000
4	Dafne	F	Tnagar	10/9/2007	7000
5	Geetha	F	Namakkal	10/9/2007	15500

//ObjectServer.java

```
D:\JDK\BIN >javac ObjectServer.java
```

```
Note: ObjectServer.java uses or overrides a deprecated API.
```

```
Note: Recompile with -Xlint:deprecation for details.
```

```
Note: ObjectServer.java uses unchecked or unsafe operations.
```

```
Note: Recompile with -Xlint:unchecked for details.
```

```
D:\JDK\BIN >java ObjectServer
```

```
Input :all
```

```
Input :5
```

//ObjectBank.java

```
D:\JDK\BIN >javac ObjectBank.java
```

//ObjectClient.java

```
D:\JDK\BIN >javac ObjectClient.java
```

```
D:\JDK\BIN >java ObjectClient
```

```
Enter the choice : (all/recordno)
```

```
all
```

```
***CUSTOMER DETAILS***
```

```
1
```

```
Anitha
```

```
F
```

```
Kodambakkam
```

```
2007-10-09 00:00:00
```

```
10000
```

CUSTOMER DETAILS

2
Banu
F
Tnagar
2007-10-09 00:00:00
15000

CUSTOMER DETAILS

3
Chitra
F
Kodambakkam
2007-10-09 00:00:00
30000

CUSTOMER DETAILS

4
Dafne
F
Tnagar
2007-10-09 00:00:00
7000

CUSTOMER DETAILS

5
Geetha
F
Namakkal
2007-10-09 00:00:00
15500
D:\JDK\BIN >java ObjectClient
Enter the choice : (all/recordno)
5

CUSTOMER DETAILS

Geetha
F
Namakkal
2007-10-09 00:00:00
15500

27. Telnet Client

```
import java.io.*;
import java.net.*;
import java.util.*;
public class telnetclient
{
    public static void main(String args[]) throws Exception
    {
        Socket s=new Socket("localhost",8080);
        DataInputStream din=new DataInputStream(s.getInputStream());
        PrintStream ps=new PrintStream(s.getOutputStream());
        DataInputStream in=new DataInputStream(System.in);
        ps.println("Telnet");
        String str=din.readLine();
        System.out.println(str);
        String str1=din.readLine();
        System.out.println(str1);
        String uname=in.readLine();
        ps.println(uname);
        String str2=din.readLine();
        System.out.println(str2);
        String pass=in.readLine();
        ps.println(pass);
        String acc=din.readLine();
        System.out.println(acc);
        String cmd=in.readLine();
        ps.println(cmd);
        if(cmd.equals("dir"))
        {
            while(true)
            {
                String rec=din.readLine();
                if(rec.equals("end"))
                    break;
                System.out.println(rec);
            }
        }
        else
        {
            String quit=din.readLine();
            System.out.println(quit);
        }
    }
}
```

28. Telnet Sever

```
import java.io.*;
import java.net.*;
import java.util.*;
public class telnets
{
    public static void main(String args[]) throws Exception
    {
        ServerSocket ss=new ServerSocket(8080);
        Socket s=ss.accept();
        ServiceClient(s);
    }
    public static void ServiceClient(Socket s)throws Exception
    {
        DataInputStream dis=null;
        PrintStream ps=null;
        try
        {
            dis=new DataInputStream(s.getInputStream());
            ps=new PrintStream(s.getOutputStream());
            String tel=dis.readLine();
            System.out.println(tel);
            ps.println("Hello Telnet Started");
            ps.println("Login : ");
            String uname=dis.readLine();
            System.out.println(uname);
            ps.println("Password : ");
            String pass=dis.readLine();
            System.out.println(pass);
            if((pass.equals("admin")) && (uname.equals("admin")))
            ps.println("Ok Server Accepted");
            while (true)
            {
                String cmd=dis.readLine();
                if(cmd.equals("quit"))
                {
                    ps.println("bye");
                    break;
                }
                else if(cmd.equals("dir"))
                {
                    File f=new File("\\");
                    String files[]=f.list();
                    for(int i=0;i<files.length;i++)
                    {
                        ps.println(files[i]);
                    }
                }
            }
        }
        catch (Exception e)
        {
            e.printStackTrace();
        }
    }
}
```

```

        }
        ps.println("end");
        break;
    }
}
}
finally
{
    ps.close();
    dis.close();
    s.close();
}
}
}
}

```

Output:

//telnetserver.java

D:\JDK\BIN >javac telnetserver.java

D:\JDK\BIN >java telnetserver

admin

//telnetclient.java

D:\JDK\BIN >javac telnetclient.java

D:\JDK\BIN >java telnetclient

Hello telnet started

Login: admin

Password: admin

dir

document and setting

winnt

jdk1.3

intel

recyeler

D:\JDK\BIN >java telnetclient

Hello telnet started

Login: admin

Password: admin

Ok. Server accepted

Quit

Bye

29. Simple Mail Transfer Protocol Server

```
import java.io.*;
import java.net.*;
class SMTPServer
{
public static void main(String args[])
{
    try
    {
        ServerSocket ss=new ServerSocket(8080);
        Socket s=ss.accept();
        ServiceClient(s);
    }
    catch(Exception e)
    {
        System.out.println("Exception caught in main:"+e);
    }
}
public static void ServiceClient(Socket s)throws Exception
{
    try
    {
        DataInputStream dis=null;
        PrintStream ps=null;
        dis=new DataInputStream(s.getInputStream());
        ps=new PrintStream(s.getOutputStream());
        FileWriter f=new FileWriter("abc.eml");
        String tel=dis.readLine();
        if(tel.equals("Ready"))
            System.out.println("Ready Signal Received from client:Client Accepted");
        ps.println("Enter the from address:");
        String from=dis.readLine();
        f.write("from:"+from+"\n");
        ps.println("Enter the to address:");
        String to=dis.readLine();
        f.write("to:"+to+"\n");
        ps.println("Enter the Message:");
        String msg=dis.readLine();        System.out.println(msg);f.write("\n");
    }
}
```

```

        f.write("Message:"+msg+"\n");
        f.close();
    }
    catch(Exception e)
    {
        System.out.println("Exception caught:"+e);
    }
}
}

```

30. Simple Mail Transfer Protocol Client

```

import java.io.*;
import java.net.*;
class SMTPClient
{
public static void main(String args[])throws Exception
{
    try
    {
        Socket s=new Socket("localhost",8080);
        DataInputStream dis=new DataInputStream(s.getInputStream());
        DataInputStream in=new DataInputStream(System.in);
        PrintStream ps=new PrintStream(s.getOutputStream());
        ps.println("Ready");
        String resp=dis.readLine();
        System.out.println("Enter the from address:");
        String strf=in.readLine();
        ps.println(strf);
        System.out.println("Enter the to address:");
        String strt=in.readLine();
        ps.println(strt);
        System.out.println("Enter the Message:");
        while(true)
        {
            String msg=in.readLine();
            ps.println(msg);
            if(msg.equals("Quit"))
            {
                System.out.println("Message is delivered to server and

```

```

        client quits");
        break;
    }
}
}
catch(Exception e)
{
    System.out.println("Exception caught:"+e);
}
}
}

```

Output:

//SMTPServer.java

D:\JDK\BIN >javac SMTPServer.java

D:\JDK\BIN >java SMTPServer

Connected

Message accepted for Delivery

D:\JDK\BIN >javac SMTPClient.java

D:\JDK\BIN >java SMTPClient

Enter From address

jeya

Enter To address

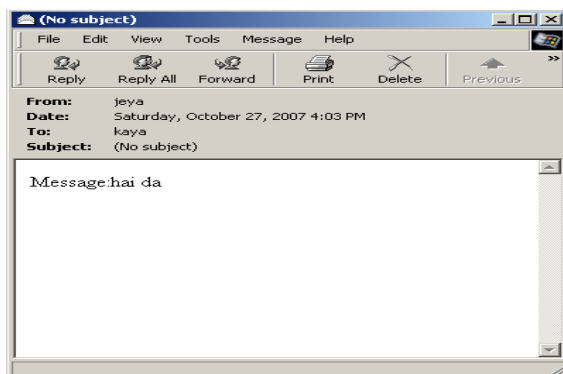
Kaya

Enter the Message

hai da

Message accepted for Delivery

//abc.eml



31. Sandard Client

```
import java.rmi.*;
import java.io.*;
import java.sql.*;
public class StdClient
{
    public static void main(String args[])
    {
        try
        {
            String Stdserverurl="rmi://" + args[0] + "/Stdserver";
            Stdinterface intf=(Stdinterface)Naming.lookup(Stdserverurl);
            Student s=null;
            System.out.println("Enter id:");
            DataInputStream dis=new DataInputStream(System.in);
            String id=dis.readLine();
            System.out.println(id);
            s=intf.getinfo(id);
            System.out.println();
            System.out.println("Student id:" + s.getid());
            System.out.println("Student Name:" + s.getname());
            System.out.println("Student Branch:" + s.getbranch());
            System.out.println("Student Grade:" + s.getgrade());
        }
        catch(Exception e)
        {
            System.out.println(e);
        }
    }
}
```

32. Standard Server

```
import java.rmi.*;
import java.net.*;
import java.sql.*;
public class Stdserver
{
```

```

public static void main(String args[])
{
    try
    {
        Stdimpl impl=new Stdimpl();
        Naming.rebind("Stdserver",impl);
    }
    catch(Exception e)
    {
        System.out.println(e);
    }
}

```

33. Standard implementation

```

import java.sql.*;
import java.rmi.*;
import java.rmi.server.*;
public class Stdimpl extends UnicastRemoteObject implements Stdinterface
{
    Connection con=null;
    Statement stmt;
    ResultSet rs;
    public Stdimpl() throws RemoteException
    {
        super();
    }
    public void connect()
    {
        try
        {
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
            con=DriverManager.getConnection("jdbc:odbc:dsnfirst");
            stmt=con.createStatement();
        }
        catch(Exception e)
        {
            System.out.println(e);
        }
    }
}

```

```

public Student getinfo(String id1)
{
    Student stu=new Student();
    try
    {
        connect();
        rs=stmt.executeQuery("select * from student where id='"+id1+"'");
        rs.next();
        stu.setid(rs.getString(1));
        stu.setname(rs.getString(2));
        stu.setbranch(rs.getString(3));
        stu.setgrade(rs.getString(4));
        con.close();
    }
    catch(Exception e)
    {
        System.out.println(e);
    }
    return stu;
}
}

```

34. Standard Interface

```

import java.rmi.*;
import java.net.*;
public interface Stdinterface extends Remote
{
    public Student getinfo(String id)throws RemoteException;
}

```

35. Implementation of Serialization

```

import java.io.*;
import java.rmi.*;
import java.net.*;
public class Student implements Serializable
{
    String id,name,branch,grade;
    public Student()

```

```

    {
        id=null;
        name=null;
        branch=null;
        grade=null;
    }
    public void setid(String idno)
    {
        id=idno;
    }
    public void setname(String n)
    {
        name=n;
    }
    public void setbranch(String br)
    {
        branch=br;
    }
    public void setgrade(String g)
    {
        grade=g;
    }
    public String getid()
    {
        return id;
    }
    public String getname()
    {
        return name;
    }
    public String getbranch()
    {
        return branch;
    }
    public String getgrade()
    {
        return grade;
    }
}

```

OUTPUT

```
D:\JDK\BIN >start rmiregistry
D:\JDK\BIN >rmic Stdimpl
D:\JDK\BIN >java Stdserver
D:\JDK\BIN >java StdClient localhost
Enter id:
1
1
Student id:1
Student Name:Asha
Student Branch:IT
Student Grade:A
```

36. Package Java Database Connectivity

```
import java.sql.*;
import java.io.*;
public class CRUD
{
    Connection con;
    PreparedStatement ps;
    public CRUD() {
        try
        {
            Class.forName("com.mysql.jdbc.Driver");
            con = DriverManager.getConnection("jdbc:mysql://localhost/test?user=root&password=root");
        }
        catch (Exception e)
        {
            e.printStackTrace();
        }
    }
    public void insertEmp(String code, String name, String city, int sal)
    {
        try
        {
            ps = con.prepareStatement("insert into Emp values(?,?,?,?)");
            ps.setString(1, code);
            ps.setString(2, name);
            ps.setString(3, city);
            ps.setInt(4, sal);
            int i = ps.executeUpdate();
        }
    }
}
```



```

    if (i != 0)
    {
        System.out.println("Inserted");
    } else
    {
        System.out.println("not Inserted");
    }
} catch (Exception e)
{
    e.printStackTrace();
}
}
}

public void updateEmp(String code, String city, int salary)
{
    try {
        ps = con.prepareStatement("update emp set city=?,salary=salary+? where code=?");
        ps.setString(1, city);
        ps.setInt(2, salary);
        ps.setString(3, code);
        int i = ps.executeUpdate();
        if (i != 0)
        {
            System.out.println("updated");
        }
        else
        {
            System.out.println("not updated");
        }
    } catch (Exception e)
    {
        e.printStackTrace();
    }
}

public void deleteEmp(String code)
{
    try
    {
        ps = con.prepareStatement("delete from emp where code=?");
        ps.setString(1, code);
        int i = ps.executeUpdate();
    }
}

```

```

if (i != 0)
{
    System.out.println("deleted");
}
else
{
    System.out.println("not deleted");
}
    }
    catch (Exception e)
    {
        e.printStackTrace();
    }
}
public void dispAll()
{
    try
    {
        Statement st = con.createStatement();
        ResultSet res = st.executeQuery("select * from emp");
        while (res.next())
        {
            System.out.print(res.getString(1));
            System.out.print(res.getString(2));
            System.out.print(res.getString(3));
            System.out.println(res.getString(4));
        }

    }
    catch (SQLException e)
    {
        e.printStackTrace();
    }
}
public void dispAnEmp(String s)
{
    try
    {
        ps = con.prepareStatement("select * from Emp where code=?");
        ps.setString(1, s);
        ResultSet res = ps.executeQuery();
        if (res.next())
        {
            System.out.print(res.getString(1));
            System.out.print(res.getString(2));
            System.out.print(res.getString(3));
            System.out.println(res.getString(4));
        }
    }
}

```

```

    } catch (SQLException e)
    {
        e.printStackTrace();
    }
}
public static void main(String[] args)
{
    try
    {
        CRUD obj = new CRUD();
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        int ch = 0;
        while (true)
        {
            System.out.println("Employee management \n" + "1. Add Employee \n " + "2. Edit Employee \n " + "3.
            Delete Employee \n " + "4. Display all record \n " + "5. Display a record \n" + "6. Exit" + "Enter option
            \n");
            String str1 = br.readLine().toString();
            ch = Integer.parseInt(str1);
            switch (ch)
            {
                case 1:
                {
                    System.out.println("Enter Employee Code");
                    String code = br.readLine();
                    System.out.println("Enter Employee Name");
                    String name = br.readLine();
                    System.out.println("Enter Employee City");
                    String city = br.readLine();
                    System.out.println("Enter Employee Salary");
                    String sal = br.readLine();
                    int salary = Integer.parseInt(sal);
                    obj.insertEmp(code, name, city, salary);
                    break;
                }
                case 2:
                {
                    System.out.println("Enter Employee Code");
                    String code = br.readLine();
                    System.out.println("Enter Employee City");
                    String city = br.readLine();
                    System.out.println("Enter Employee Salary");
                    String sal = br.readLine();
                    int salary = Integer.parseInt(sal);
                    obj.updateEmp(code, city, salary); break;
                }
            }
        }
    }
}

```

```

case 3:
    {
        System.out.println("Enter Employee Code to delete record");
        String code = br.readLine();
        obj.deleteEmp(code);
        break;
    }
case 4:
    {
        obj.dispAll();
        break;
    }
case 5:
    {
        System.out.println("Enter Employee Code to display record");
        String code = br.readLine();
        obj.dispAnEmp(code);
        break;
    }
case 6:
    {
        System.exit(0);
    }
default:
break;
    }
} catch (Exception e)
    {
        e.printStackTrace();
    }
}
}

```

37. Using Menu's and Frame

```

import java.awt.*;
class AWTMenu extends Frame
{
    MenuBar mbar;
    Menu menu,submenu;
    MenuItem m1,m2,m3,m4,m5;
    public AWTMenu()
    {

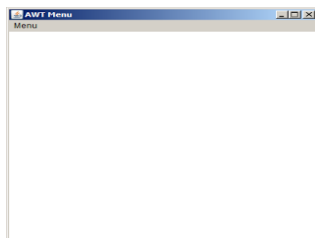
```

```

// Set frame properties
setTitle("AWT Menu"); // Set the title
setSize(400,400); // Set size to the frame
setLayout(new FlowLayout()); // Set the layout
setVisible(true); // Make the frame visible
setLocationRelativeTo(null); // Center the frame
// Create the menu bar
mbar=new MenuBar();
// Create the menu
menu=new Menu("Menu");
// Create the submenu
submenu=new Menu("Sub Menu");
// Create MenuItems
m1=new MenuItem("Menu Item 1");
m2=new MenuItem("Menu Item 2");
m3=new MenuItem("Menu Item 3");
m4=new MenuItem("Menu Item 4");
m5=new MenuItem("Menu Item 5");
// Attach menu items to menu
menu.add(m1);
menu.add(m2);
menu.add(m3);
// Attach menu items to submenu
submenu.add(m4);
submenu.add(m5);
// Attach submenu to menu
menu.add(submenu);
// Attach menu to menu bar
mbar.add(menu);
// Set menu bar to the frame
setMenuBar(mbar);
}
public static void main(String args[])
{
new AWTMenu();
}
}

```

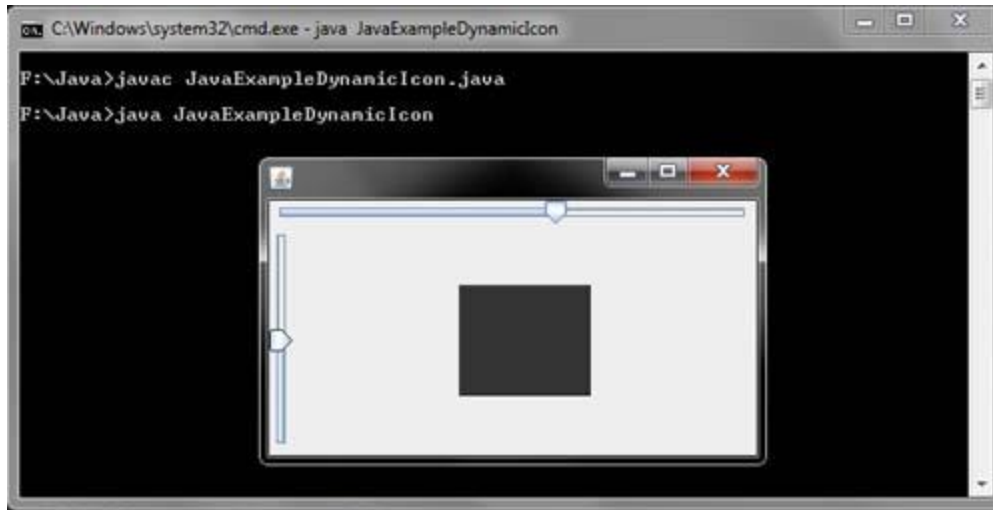
AWTMenu() : Code for creation of menus is written here
new AWTMenu() : Create the object for the class



38. The creation and display of the JSlider control is explained in Program.

```
import javax.swing.*;
import javax.swing.event.*;
import java.awt.*;
public class JavaExampleDynamicIcon
{
    public static void main(String[] as)
    {
        final JSliderWdth = new JSlider(JSlider.HORIZONTAL, 1, 150, 75);
        final JSliderHght = new JSlider(JSlider.VERTICAL, 1, 150, 75);
        class DynamicIcon implements Icon
        {
            public intgetIconWidth()
            {
                return Wdth.getValue();
            }
            public intgetIconHeight()
            {
                return Hght.getValue();
            }
            public void paintIcon(Component Cmpnt, Graphics gr, int a, int b)
            {
                gr.fill3DRect(a, b, getIconWidth(), getIconHeight(), true);
            }
        };
        Icon Icn = new DynamicIcon();
        final JLabelLblDynmc = new JLabel(Icn);
        class Updater implements ChangeListener
        {
            public void stateChanged(ChangeEventEvnt)
            {
                LblDynmc.repaint();
            }
        };
        Updater Updtr = new Updater();
        Wdth.addChangeListener(Updtr);
        Hght.addChangeListener(Updtr);
        JFrameFrm = new JFrame();
        Container Cntnr = Frm.getContentPane();
        Cntnr.setLayout(new BorderLayout());
        Cntnr.add(Wdth, BorderLayout.NORTH);
        Cntnr.add(Hght, BorderLayout.WEST);
        Cntnr.add(LblDynmc, BorderLayout.CENTER);
        Frm.setSize(210,210);
        Frm.setVisible(true);
    }
}
```

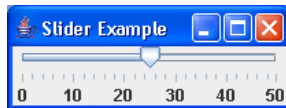
OUTPUT:



39. Swing Slider

```
import java.awt.BorderLayout;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.JSlider;
public class SwingSliderExample extends JPanel {
public SwingSliderExample() {
super(true);
this.setLayout(new BorderLayout());
JSlider slider = new JSlider(JSlider.HORIZONTAL, 0, 50, 25);
slider.setMinorTickSpacing(2);
slider.setMajorTickSpacing(10);
slider.setPaintTicks(true);
slider.setPaintLabels(true);
slider.setLabelTable(slider.createStandardLabels(10));
add(slider, BorderLayout.CENTER);
}
public static void main(String s[]) {
JFrame frame = new JFrame("Slider Example");
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame.setContentPane(new SwingSliderExample());
frame.pack();
frame.setVisible(true);
}
}
```

OUTPUT:



40. Swing -Standard Color

```
import java.awt.Color;
import java.awt.Dimension;
import java.awt.EventQueue;
import java.util.ArrayList;
import java.util.List;
import javax.swing.GroupLayout;
import javax.swing.JFrame;
import static javax.swing.JFrame.EXIT_ON_CLOSE;
import javax.swing.JLabel;
import javax.swing.JPanel;
class MyLabel extends JLabel {
public MyLabel() {
        super("", null, LEADING);
    }
public boolean isOpaque() {
return true;
    }
}
public class StandardColoursEx extends JFrame {

public StandardColoursEx() {
        initUI();
    }

private void initUI()
{
Color[] stdCols = {Color.black, Color.blue, Color.cyan, Color.darkGray, Color.gray, Color.green,
Color.lightGray,Color.magenta, Color.orange, Color.pink, Color.red, Color.white, Color.yellow};
List<JLabel> labels = new ArrayList();
for (Color col : stdCols) {
JLabel lbl = createColouredLabel(col);
labels.add(lbl);
}
createLayout(labels.toArray(new JLabel[labels.size()]));
setTitle("Standard colours");
setLocationRelativeTo(null);
setDefaultCloseOperation(EXIT_ON_CLOSE);
}
public JLabel createColouredLabel(Color col) {
MyLabel lbl = new MyLabel();
lbl.setMinimumSize(new Dimension(90, 40));
lbl.setBackground(col);
return lbl;
}
private void createLayout(JLabel[] labels) {
```



```

JPanel pane = (JPanel) getContentPane();
GroupLayout gl = new GroupLayout(pane);
pane.setLayout(gl);
pane.setToolTipText("Content pane");
gl.setAutoCreateContainerGaps(true);
gl.setAutoCreateGaps(true);
gl.setHorizontalGroup(gl.createParallelGroup()
    .addGroup(gl.createSequentialGroup()
        .addComponent(labels[0])
        .addComponent(labels[1])
        .addComponent(labels[2])
        .addComponent(labels[3]))
    .addGroup(gl.createSequentialGroup()
        .addComponent(labels[4])
        .addComponent(labels[5])
        .addComponent(labels[6])
        .addComponent(labels[7]))
    .addGroup(gl.createSequentialGroup()
        .addComponent(labels[8])
        .addComponent(labels[9])
        .addComponent(labels[10])
        .addComponent(labels[11]))
    .addComponent(labels[12])
);
gl.setVerticalGroup(gl.createSequentialGroup()
    .addGroup(gl.createParallelGroup()
        .addComponent(labels[0])
        .addComponent(labels[1])
        .addComponent(labels[2])
        .addComponent(labels[3]))
    .addGroup(gl.createParallelGroup()
        .addComponent(labels[4])
        .addComponent(labels[5])
        .addComponent(labels[6])
        .addComponent(labels[7]))
    .addGroup(gl.createParallelGroup()
        .addComponent(labels[8])
        .addComponent(labels[9])
        .addComponent(labels[10])
        .addComponent(labels[11]))
    .addComponent(labels[12])
);
pack();
}

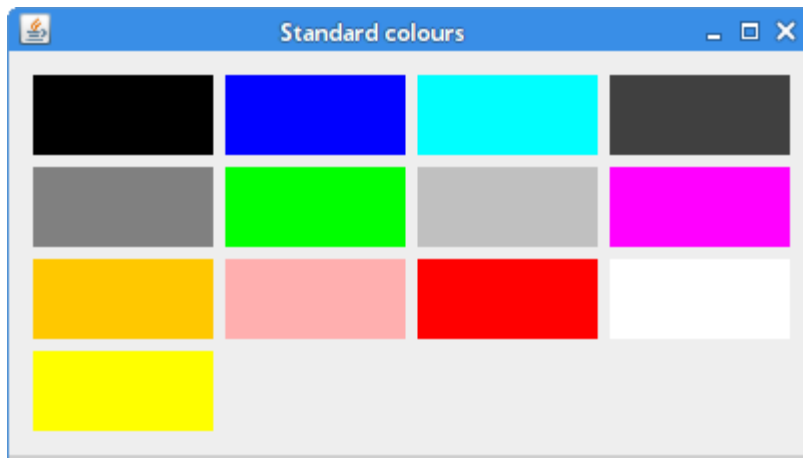
```

```

public static void main(String[] args) {

EventQueue.invokeLater(() -> {
StandardColoursEx ex = new StandardColoursEx();
ex.setVisible(true);
});
}
}

```



41. Creating Login page

```

package com.zetcode;
import java.awt.Component;
import java.awt.Container;
import java.awt.EventQueue;
import java.awt.event.ActionEvent;
import java.util.Arrays;
import javax.swing.AbstractAction;
import javax.swing.GroupLayout;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPasswordField;
import javax.swing.JTextField;
import static javax.swing.LayoutStyle.ComponentPlacement.UNRELATED;
public class PasswordEx extends JFrame {
    private JTextFieldloginField;
    private JPasswordFieldpassField;
    public PasswordEx() {
        initUI();
    }
    private void initUI() {

```

```

JLabel lbl1 = new JLabel("Login");
JLabel lbl2 = new JLabel("Password");
loginField = new JTextField(15);
passField = new JPasswordField(15);
JButton submitButton = new JButton("Submit");
submitButton.addActionListener(new SubmitAction());
createLayout(lbl1, loginField, lbl2, passField, submitButton);
setTitle("Login");
setLocationRelativeTo(null);
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}
private class SubmitAction extends AbstractAction {
public void actionPerformed(ActionEvent e) {
doSubmitAction();
}
private void doSubmitAction() {
String login = loginField.getText();
char[] passwd = passField.getPassword();
if (!login.isEmpty() && passwd.length != 0) {
System.out.format("User %s entered %s password%n",
login, String.valueOf(passwd));
}
Arrays.fill(passwd, '0');
}
}
private void createLayout(Component... arg) {
Container pane = getContentPane();
GroupLayout gl = new GroupLayout(pane);
pane.setLayout(gl);
gl.setAutoCreateGaps(true);
gl.setAutoCreateContainerGaps(true);
gl.setHorizontalGroup(gl.createSequentialGroup()
.addGap (50)
.addGroup (gl.createParallelGroup()
.addComponent(arg[0])
.addComponent(arg[1])
.addComponent(arg[2])
.addComponent(arg[3])
.addComponent(arg[4]))
.addGap(50) );
}

```

```

gl.setVerticalGroup(gl.createSequentialGroup()
    .addGap(50)
    .addGroup(gl.createSequentialGroup()
        .addComponent(arg[0])
        .addComponent(arg[1], GroupLayout.DEFAULT_SIZE,
GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)
        .addComponent(arg[2])
        .addComponent(arg[3], GroupLayout.DEFAULT_SIZE,
GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(UNRELATED)
        .addComponent(arg[4]))
    .addGap(50)
);
pack();
}
public static void main(String[] args) {
EventQueue.invokeLater() -> {
PasswordEx ex = new PasswordEx();
ex.setVisible(true);
});
}
}

```

Output:

