Java and J2EE Lab

Course Code : IS624L **Credits : 0:0:1**

Prerequisites: IS315 (OOPS) **Contact Hours : 28 hrs**

Course coordinator(s): Sumana M

Course Objectives:

1. Design java applications and be able to implement them.
2. Understand the importance of java concepts and their use in real world applications.
3. Ability to choose appropriate java concepts for problem solving.

Syllabus

1. **Create a class called account with the data members(Accno – integer, name String, Phone_No: integer, balance_amt:float), and following methods :**
    a. **getinput() to get input from the user**
    b. **Deposit() method which takes the amount to be deposited in to his/her account and do the calculation.**
    c. **Withdraw() method which gets the amount to be withdrawn from his/her account.**

    Print the appropriate results.

```
 public class Account
 { String name;
   double balance;
   public getInput  (String n, double b)

{ name = n;
   balance = b; }// end Account

public double deposit(double d)
{
if ( d >= 0)
balance = balance + d;
if (d < 0)
System.out.println("Your deposit has to be >= 0");
return balance;
}
public double withdrawl(double w)
 {
 if (w >= 0 && w <= balance)
 balance = balance -w;
if (w > balance)
System.out.println("You don't have sufficient funds!");
if (w < 0)
System.out.println("Withdraw must be >= 0");
 return balance; }
}
```

2. Write a Java Program that does the following related to Inheritance:

a. **Create an abstract class called Vehicle which contains the ‘year_of_manufacture’ data member and two abstract methods ‘getData()’ and ‘putData()’ with a constructor.**
b. **Create two derived classes “TwoWheeler” and “FourWheeler” and implement the abstract methods. Make “FourWheeler” as final class.**
c. **Create class ‘MyTwoWheeler’ which is a sub-class of “TwoWheeler” and demonstrate the use of super keyword to initialize data members of “MyTwoWheeler”.**

```
import java.util.*;

abstract class Vehicle
{
        int year_of_manfacture;
        public abstract void getData();
        public abstract void putData();
}

class TwoWheeler extends Vehicle
{
        protected String brand;
        protected int cost;
        String EngineType;
        public String color;
        Scanner s;
        public void getData()
        {

                        s = new Scanner(System.in);
                         System.out.println("Enter Brand of TwoWheeler:");
                         brand=s.next();
                        System.out.println("Enter Cost of TwoWheeler");
                        cost = Integer.parseInt(s.next());
                        System.out.println("Enter Engine Type TwoWheeler (2_Stroke or 4_ Stroke)");
                        EngineType = s.next();
                        System.out.println("Enter Color of TwoWheeler");
                        color = s.next();
        }
```

```
    public void putData()
    {
        System.out.println("TWO WHEELER\nBrand:" + brand + "\nEngineType: "
                + EngineType + "\nCost:" + cost + "\nColor:" + color+"\n\n");
    }
}

final class FourWheeler extends Vehicle
{
    private String brand;
    protected int cost;
    String EngineType;
    public String color;
    Scanner s;
    public void getData()
    {
            s = new Scanner(System.in);
             System.out.println("Enter Brand of FourWheeler:");
             brand=s.next();
            System.out.println("Enter Cost of FourWheeler");
            cost = Integer.parseInt(s.next());
            System.out.println("Enter Engine Type of FourWheeler (2_Stroke or 4_Stroke)");
            EngineType = s.next();
            System.out.println("Enter Color of FourWheeler");
            color = s.next();
    }
    public void putData()
    {
        System.out.println("FOUR WHEELER\nBrand:" + brand + "\nEngineType: "
                + EngineType + "\nCost:" + cost + "\nColor:" + color+"\n\n");
    }
}
class MyTwoWheeler extends TwoWheeler
{
    String OwnerName;
    MyTwoWheeler(String ow,String br, int co, String et, String colo)
    {
        OwnerName = ow;
        super.brand=br;
        super.cost = co;
        super.EngineType = et;
        super.color = colo;
    }

    public void prin()
    {
        System.out.println("The Vehicle \nBrand:"+super.brand+"\nEngineType:" +
super.EngineType + "\nCost:" + super.cost + "\nColor: "  + super.color);
```

```
                System.out.println("Owner:" + OwnerName);
        }
}

class first
{
        public static void main(String args[])
        {
                TwoWheeler tw = new TwoWheeler();
                tw.getData();
                tw.putData();
                FourWheeler fw=new FourWheeler();
                fw.getData();
                fw.putData();
                MyTwoWheeler mt = new MyTwoWheeler("Ramesh","Bajaj",50000, "2 Stroke", "Red");
                mt.prin();
        }
}
```

Output:

```
Enter Brand of TwoWheeler:
Tvs
Enter Cost of TwoWheeler
33000
Enter Engine Type TwoWheeler (2_Stroke or 4_Stroke)
2_Stroke
Enter Color of TwoWheeler
Red
TWO WHEELER
Brand:Tvs
EngineType: 2_Stroke
Cost:33000
Color:Red
Enter Brand of FourWheeler:
Bajaj
Enter Cost of FourWheeler
40000
Enter Engine Type of FourWheeler (2_Stroke or 4_Stroke)
4_Stroke
Enter Color of FourWheeler
Blue
FOUR WHEELER
Brand:Bajaj
EngineType: 4_Stroke
Cost:40000
Color:Blue
The Vehicle
Brand:Bajaj
```

```
EngineType:2 Stroke
Cost:50000
Color: Red
Owner:Ramesh
```

3. Write a Java Program that does the following related to Packages and Interfaces , Exception Handling:

a. Create an interface Student which gets the name and branch of a student.

b. Create a package called ‘StudentPackage’ which has a user-defined class RegisterStudent.

c. If a student registers above 30 credits for the semester, the method should throw a user-defined exception called ‘*CreditLimit*’ and display an appropriate message.

d. Create another package called ‘ResultPackage’ which displays the grade for the subject registered for particular semester and if the CGPA is in invalid format throw NumberFormatException also if CGPA is above 10 then throws an InvalidCGPA user-defined exception.

e. Collect the marks of all the semesters and display the SGPA with minimum calculation of 4 semesters.

StudentPackage:

```
import java.util.*;

class RegisterStudent
{

   void total(int cd)
   {
           int TotalCredits=cd;
          try
                    {
                             if(TotalCredits>30)
                       {
                             throw new CreditLimit(TotalCredits);
                       }
                             else
                             {
                                       System.out.println("The Total credit is:"+TotalCredits);
                             }
                    }
                    catch(Exception e)
                    {
                        System.out.println("Total Credits is more than 30");
                }
    }
}
class CreditLimit extends Exception
{
        CreditLimit(int i)
  {
          System.out.println("The Total credit is:"+i);
  }
}
public class student {

        static Scanner s;
        public static void main(String[] args)
        {
                s = new Scanner(System.in);
                System.out.println("Enter the total credit:");
                int c=Integer.parseInt(s.next());
                RegisterStudent  r=new RegisterStudent ();
                 r.total(c);
        }
}
```

ResultPackage:

```
import java.io.*;
import java.util.Scanner;
class grade
{
        String sub1,sub2,sub3;
        Scanner s=new Scanner(System.in);
        void getGrade()
        {
                 sub1="A";
                 sub2="A";
                 sub3="S";
        }
        void display()
        {
                System.out.println("Grade of Registered subject");
                System.out.println(" Grade of subject1:"+sub1);
                System.out.println(" Grade of subject2:"+sub2);
                System.out.println(" Grade of subject3:"+sub3);
        }
        void cgpa()
        {
                 double input;
                 Scanner scan = new Scanner(System.in);
             System.out.println("Enter an CGPA: ");
             try
             {
                 input = Double.parseDouble(scan.next());
                 if(input > 10)
                 System.out.println("Not a valid CGPA!!");
                else
                System.out.println("CGPA is: " + input);
             }
             catch (NumberFormatException e)
             {
                System.out.println("Not a valid CGPA!!");
             }
        }
}
public class Result {

        public static void main(String[] args)
        {
    grade g=new grade();
    g.getGrade();
    g.display();
    g.cgpa();
        }

}
```

Output:

StudentPackage:

Case1:

```
Enter the total credit:
27
The Total credit is:27
```

Case2:

```
Enter the total credit:
32
The Total credit is:32
Total Credits is more than 30
```

ResultPackage:

Case1:

```
Grade of Registered subject
 Grade of subject1:A
 Grade of subject2:A
 Grade of subject3:S
Enter an CGPA:
9.47
CGPA is: 9.47
```

Case2:

```
Grade of Registered subject
 Grade of subject1:A
 Grade of subject2:A
 Grade of subject3:S
Enter an CGPA:
nine
Not a valid CGPA!!
```

4. Write a Program that simulates a telephone that records missed incoming calls. For each missed call, store the time of call, telephone number of origin, and name of the called if the name is available. For unlisted numbers, set the name to “private caller”. Choose or extend the most appropriate collection class and provide the following features.

- **a) Numbers are recalled in the order they arrive**
- **b) Up to 10 numbers are recorded. When the eleventh call comes in, it is stored and the oldest call is deleted so that no more than 10 numbers are ever recorded.**
- **c) After each number display, the user can select**
  - **a. to delete the call**
  - **b. to go on to the next missed call, or**
  - **c. to display the call details (number, caller name and time).**
- **d) Delete the number if user specifies a number to delete.**

Write a helper class to represent an incoming call with fields to hold the number, name of the caller, and time of the call. Write a tester call that stores the several numbers, simulate the user pressing the missed-calls button, and finally prints the entire collection of stored calls.

```
import java.util.Calendar;
import java.util.HashMap;
import java.util.Map;
import java.util.Random;
import java.util.Scanner;
import java.util.Set;
class phone
{
    String name, time;
    String phnumber;
    phone(String na, String n, String t)
    {
        name = na;
        phnumber = n;
        time = t;
    }
    public String toString()
    {
        return "Name is:" + name + " Number: " + phnumber + " Time: " + time;
    }
}

class PhoneNumber
{
    private static Scanner s;
    public static void main(String args[])
    {
        String[] lot = new String[10];
        int ii = 0;
        HashMap<Integer, phone> pm = new HashMap<Integer, phone>();
        HashMap<Integer, String> nm = new HashMap<Integer, String>();
        nm.put(1, "Ram");
        nm.put(2, "Sam");
        nm.put(3, "Wong");
        nm.put(4, "john");
        nm.put(5, "Ravi");
        nm.put(6, "Rajesh");
        nm.put(7, "Ramesh");
        nm.put(8, "Private");
        nm.put(9, "Private");
        nm.put(10, "Private");
        HashMap<Integer, String> sm = new HashMap<Integer, String>();
        sm.put(1, "122345");
        sm.put(2, "234552");
        sm.put(3, "252452");
        sm.put(4, "525551");
        sm.put(5, "524255");
        sm.put(6, "525245");
        sm.put(7, "253452");
        sm.put(8, "523455");
        sm.put(9, "5325275");
        sm.put(10, "253565");
        Calendar c = Calendar.getInstance();
        s = new Scanner(System.in);
        int count = 1;
        while (true)
        {
```

```
System.out.println("Enter the choice\n 1)add 2)Disply 3)Delete 4)exit");
int ch = Integer.parseInt(s.next());
switch (ch)
{
case 1:
	if (count <= 10)
	{
		Random rand = new Random();
		int number = rand.nextInt(10);
		String Name = nm.get(number);
		String Number = sm.get(number);
		int hour = c.get(Calendar.HOUR);
		int min = c.get(Calendar.MINUTE);
		int sec = c.get(Calendar.SECOND);
		String time = hour + ":" + min + ":" + sec;
		pm.put(count, new phone(Name, Number, time));
		count++;
	}
	else
		pm.remove(1);
	break;
case 2:
	Set<Map.Entry<Integer, phone>> myset = pm.entrySet();
	for (Map.Entry<Integer, phone> me : myset)
	{
		System.out.println("----------------------------\n");
		System.out.println(me.getValue());
		System.out.println("Enter the choice 1)Display next number\n 2)Delete the displayed number and print the next number");
		int cd = Integer.parseInt(s.next());
		switch (cd)
		{
		case 1:
			continue;
		case 2:
			lot[ii] = Integer.toString(me.getKey());
			ii++;
			System.out.println("Deleted!!!");
			continue;
		}
		System.out.println("\n");
	}
	for (int r = 0; r < ii; r++)
	{
		int mm = Integer.parseInt(lot[r]);
```

```
                  pm.remove(mm);
              }
              ii = 0;
              break;
          case 3:
              System.out.println("Enter the phone number to be delete");
              Scanner s1=new Scanner(System.in);
              String s3=s1.next();
              Set<Map.Entry<Integer, phone>> myset1 = pm.entrySet();
              for (Map.Entry<Integer, phone> me1 : myset1)
                  {
                  phone p1=me1.getValue();
                  String s2=p1.phnumber;
                  if(s3.equals(s2))
                  {
                      int t=me1.getKey();
                      pm.remove(t);
                      System.out.println("deleted!!");
                      break;
                  }
              }
                 break;
          case 4:System.exit(0);

          }
}}}
```

Output:

```
Enter the choice
 1)add 2)Disply 3)Delete 4)exit
1
Enter the choice
 1)add 2)Disply 3)Delete 4)exit
1
Enter the choice
 1)add 2)Disply 3)Delete 4)exit
1
Enter the choice
 1)add 2)Disply 3)Delete 4)exit
2
----------------------------
Name is:Rajesh Number: 525245 Time: 8:26:9
Enter the choice 1)Display next number
 2)Delete the displayed number and print the next number
1
```

```
----------------------------
Name is:Ram Number: 122345 Time: 8:26:9
Enter the choice 1)Display next number
 2)Delete the displayed number and print the next number
1
----------------------------
Name is:Ravi Number: 524255 Time: 8:26:9
Enter the choice 1)Display next number
 2)Delete the displayed number and print the next number
1
Enter the choice
 1)add 2)Disply 3)Delete 4)exit
2
----------------------------
Name is:Rajesh Number: 525245 Time: 8:26:9
Enter the choice 1)Display next number
 2)Delete the displayed number and print the next number
1
----------------------------
Name is:Ram Number: 122345 Time: 8:26:9
Enter the choice 1)Display next number
 2)Delete the displayed number and print the next number
2
Deleted!!!
----------------------------
Name is:Ravi Number: 524255 Time: 8:26:9
Enter the choice 1)Display next number
 2)Delete the displayed number and print the next number
1
Enter the choice
 1)add 2)Disply 3)Delete 4)exit
2
----------------------------
Name is:Rajesh Number: 525245 Time: 8:26:9
Enter the choice 1)Display next number
 2)Delete the displayed number and print the next number
1
----------------------------
Enter the choice
 1)add 2)Disply 3)Delete 4)exit
4
```

5. Write a program that uses Java Swing and JDBC to create a stand-alone application:

a. **Create two tables namely, Representative (RepNo, RepName, State, Comission, Rate) and Customer (CustNo, CustName, State, Credit_Limit, RepNo) in MySQL database.**
b. **Use appropriate Swing components to insert values in a form.**
c. **Use another form to display Representative’s information whose Credit_Limit is above 15,000.**

```
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import javax.swing.BoxLayout;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JTextArea;
import javax.swing.JTextField;
import com.mysql.jdbc.Statement;

public class myFrame extends JFrame
{

        myFrame()
        {
                super("My Jframe Example");
                JLabel jlrep = new JLabel("Representative Information");
                JLabel jl11 = new JLabel("Enter RepNo");
                final JTextField jtf11 = new JTextField();
                JLabel jl12 = new JLabel("Enter RepName");
                final JTextField jtf12 = new JTextField();
                JLabel jl13 = new JLabel("Enter State");
                final JTextField jtf13 = new JTextField();
                JLabel jl14 = new JLabel("Enter Commission");
                final JTextField jtf14 = new JTextField();
                JLabel jl15 = new JLabel("Enter Rate");
                final JTextField jtf15 = new JTextField();
                JButton jb1 = new JButton("Submit");

                JLabel jlcus = new JLabel("Customer Information");
                JLabel jl21 = new JLabel("Enter CustomerNo");
                final JTextField jtf21 = new JTextField();
                JLabel jl22 = new JLabel("Enter CustomerName");
                final JTextField jtf22 = new JTextField();
                JLabel jl23 = new JLabel("Enter State");
                final JTextField jtf23 = new JTextField();
                JLabel jl24 = new JLabel("Enter Credit limit");
                final JTextField jtf24 = new JTextField();
                JLabel jl25 = new JLabel("Enter RepNo");
                final JTextField jtf25 = new JTextField();
                JButton jb2 = new JButton("Submit");

                JPanel panel = new JPanel();
                final JTextArea jta = new JTextArea();
                jta.setRows(10);
                jta.setColumns(5);
                JButton jb3 = new JButton("click");

                panel.add(jl11);
                panel.add(jtf11);
                panel.add(jl12);
                panel.add(jtf12);
                panel.add(jl13);
                panel.add(jtf13);
                panel.add(jl14);
                panel.add(jtf14);
                panel.add(jl15);
                panel.add(jtf15);
                panel.add(jb1);

                panel.add(jl21);
                panel.add(jtf21);
                panel.add(jl22);
                panel.add(jtf22);
                panel.add(jl23);
                panel.add(jtf23);
                panel.add(jl24);
                panel.add(jtf24);
                panel.add(jl25);
                panel.add(jtf25);
                panel.add(jb2);

                panel.add(jta);
                panel.add(jb3);
                jb1.addActionListener(new ActionListener()
                {
                    public void actionPerformed(ActionEvent e)
                    {
                        try
                        {
                            Statement stmt;
                            Class.forName("com.mysql.jdbc.Driver");
                            Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/test", "root", "deek");
```

```
                    if (conn != null)
                    {
                        System.out.println("Connection successful !!!");
                        String Repno = jtf11.getText();
                        String Repname = jtf12.getText();
                        String state = jtf13.getText();
                        String commission = jtf14.getText();
                        String rate = jtf15.getText();
                        stmt = (Statement) conn.createStatement();
                    System.out.println(Repno + Repname + state + commission);
                        String query1 = "insert into Representative values('"
                                + Repno + "','" + Repname + "','" + state
                                + "','" + commission + "','" + rate + "');";
                        stmt.executeUpdate(query1);
                    }
                    else
                        System.out.println("Connection  not successful !!!");
                }
                catch (SQLException ex)
                {
                    System.out.println(ex.getMessage());
                }
                catch (ClassNotFoundException exx)
                {
                    System.out.println(exx.getMessage());
                }
            }
        });
        jb2.addActionListener(new ActionListener()
        {

            public void actionPerformed(ActionEvent e)
            {
                try
                {
                    Statement stmt2;
                    Class.forName("com.mysql.jdbc.Driver");
Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/test", "root", "deek");
                    if (conn != null)
                    {
                        System.out.println("Connection successful !!!");
                        String Custno = jtf21.getText();
                        String CustName = jtf22.getText();
                        String state = jtf23.getText();
                        String Credit = jtf24.getText();
                        int cr = Integer.parseInt(Credit);
                        String Rno = jtf25.getText();
                        stmt2 = (Statement) conn.createStatement();
                    System.out.println(Custno + CustName + state + cr + Rno);
                String query2 = "insert into Customer values('"+ Custno + "','" +
CustName +    "','" + state+ "','" + cr + "','" + Rno + "');";
                        stmt2.executeUpdate(query2);

                    }
                    else
                        System.out.println("Connection  not successful !!!");
                }
                catch (SQLException ex)
                {
                    System.out.println(ex.getMessage());
                }
                catch (ClassNotFoundException exx)
                {
                    System.out.println(exx.getMessage());
                }
            }

        });

        jb3.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                try {
                    Statement stmt;
                    Class.forName("com.mysql.jdbc.Driver");
                    Connection conn = DriverManager.getConnection(
                        "jdbc:mysql://localhost:3306/test", "root", "deek");
                    if (conn != null)
                    {
                        stmt = (Statement) conn.createStatement();
                        String query3="SELECT * FROM Representative
WHERE RepNo IN (SELECT RepNo FROM Customer WHERE Credit_Limit > 15000 )";
                        ResultSet rs = stmt.executeQuery(query3);
                        while (rs.next())
                        {
                            jta.append("Representative Information");
                            jta.append("\n");
                            jta.append("Number:");
                            jta.append(rs.getString("RepNo"));
                            jta.append("\n");
                            jta.append("Name:");
                            jta.append(rs.getString("RepName"));
                            jta.append("\n");
                            jta.append("State:");
                            jta.append(rs.getString("State"));
                            jta.append("\n");
                            jta.append("Comission:");
                            jta.append(rs.getString("Comission"));
                            jta.append("\n");
                            jta.append("Rate:");
                            jta.append(rs.getString("Rate"));
                            jta.append("\n");
                        }
                        System.out.println("Connection successful !!!");

                    }
                    else
                        System.out.println("Connection  not successful !!!");
                }
                catch (SQLException ex)
                {
                    System.out.println(ex.getMessage());
                }
                catch (ClassNotFoundException exx)
                {
                    System.out.println(exx.getMessage());
                }
            }
        });
        setContentPane(panel);

    }

    public static void main(String[] args) {
        myFrame mf = new myFrame();
        mf.getContentPane().setLayout(
                new BoxLayout(mf.getContentPane(), BoxLayout.Y_AXIS));
        mf.setVisible(true);
        mf.setDefaultCloseOperation(EXIT_ON_CLOSE);
        mf.pack();
    }
}
```

6. Write a Java program to create a progress bar GUI which displays the copy status of writing contents of one file to another file. Terminate the progress bar on click of STOP button and display the contents of the file.

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.border.*;
import java.io.*;

public class ProgressBar extends JFrame implements ActionListener
{
```

```
private JProgressBar jp = new JProgressBar();
private JButton jb = new JButton("Copy");
private JButton jc = new JButton("Cancel");
private JTextField fromFile = new JTextField();
private JTextField toFile = new JTextField();
private ProgressBarThread thread;

public static void main(String[] args)  // creating the frame application
{
  ProgressBar application = new ProgressBar();
  application.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
  application.setTitle("Copy file");
  application.setSize(400, 180);
  application.setVisible(true);
}

public ProgressBar()                    // class constructor
{
  Container pane = getContentPane();    // building the GUI
  pane.setLayout(new BorderLayout());
  jp.setStringPainted(true);            // display style of the progress bar
  pane.add(jp, BorderLayout.NORTH);

  JPanel panel1 = new JPanel(new BorderLayout());
  panel1.setBorder(new TitledBorder("From:"));
  panel1.add(fromFile, BorderLayout.CENTER);

  JPanel panel2 = new JPanel(new BorderLayout());
  panel2.setBorder(new TitledBorder("To:"));
  panel2.add(toFile, BorderLayout.CENTER);

  JPanel panel3 = new JPanel(new GridLayout(2,1));
  panel3.add(panel1);
  panel3.add(panel2);
  pane.add(panel3, BorderLayout.CENTER);

  JPanel panel4 = new JPanel();
  panel4.add(jb);
  panel4.add(jc);
  pane.add(panel4, BorderLayout.SOUTH);

  jb.addActionListener(this);
  jc.addActionListener(this);
  jc.setEnabled(false);
}

public void actionPerformed(ActionEvent e) // buttons events processing
{
    if (e.getSource() == jb)
    {
      thread = new ProgressBarThread(this, jp, fromFile, toFile);
      thread.start();
      jb.setEnabled(false);
      jc.setEnabled(true);
    }
    else if (e.getSource() == jc)
    {
      thread.cancel = true;
      jc.setEnabled(false);
    }
  }
}

class ProgressBarThread extends Thread
{
  private JFrame frame;
  private JProgressBar jp;
  private JTextField fromFile;
  private JTextField toFile;
  public boolean cancel = false;

  public ProgressBarThread(JFrame f, JProgressBar jp,
                           JTextField from, JTextField to)
  {
    frame = f;                              // thread constructor
    this.jp = jp;
    fromFile = from;
    toFile = to;
  }

  public void run()
  {
    BufferedInputStream in = null;
    BufferedOutputStream out = null;
    try
    {
      File inFile = new File(fromFile.getText().trim());  // input stream
      in = new BufferedInputStream(new FileInputStream(inFile));

      File outFile = new File(toFile.getText().trim());   // output stream
      out =  new BufferedOutputStream(new FileOutputStream(outFile));

      long fileSize = in.available();        // input file size in bytes
      jp.setValue(0);                        // set up the progress bar
      jp.setMaximum(100);                    // indicator
```

```
   int r = 0;
   long bytesRead = 0;
   byte[] buffer = new byte[5];        // read/write buffer
   while ((r = in.read(buffer, 0, buffer.length)) != -1)
   {
    out.write(buffer, 0, r);         // write to the file
    bytesRead += r;
    int copyProgress = (int) (bytesRead*100.0/fileSize);
    jp.setValue(copyProgress);        // update the indicator
    if (cancel) return;            // kill the thread
   }
  }
  catch(FileNotFoundException e)
  {
   JOptionPane.showMessageDialog(frame, "File not found",
   "Error message", JOptionPane.ERROR_MESSAGE);
  }
  catch(IOException e)
  {
   JOptionPane.showMessageDialog(frame, "Cannot write to file",
   "Error message", JOptionPane.ERROR_MESSAGE);
  }
  finally                    // close the files
  {
   try
   {
    if (in != null) in.close();
    if (out != null) out.close();
   }
   catch(Exception e)
   {
    JOptionPane.showMessageDialog(frame, "I/O error",
    "Error message", JOptionPane.ERROR_MESSAGE);
   }
  }
 }
}
```

7. Create a Servlet to file IT returns that accepts personal information, salary information and Tax deduction details from the user and write the information into a file. Also accept the name of the person and display in on the page.

```
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class ITreturns extends HttpServlet {
    private static final long serialVersionUID = 1L;

    public ITreturns() {
        super();
    }
        protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        String name=request.getParameter("name");
        String gender=request.getParameter("gender");
        String salary=request.getParameter("salary");
        String tax=request.getParameter("tax");
        PrintWriter out=response.getWriter();
        File file = new File("/home/mahen/1.txt");
        file.createNewFile();
        FileOutputStream fout = new FileOutputStream(file);
        out.println(""+name+gender+salary+tax);
        fout.write(("hello"+name+gender+salary+tax).getBytes());
        fout.close();

    }
       protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        }

}
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
info.jsp
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
<body>
<form action="ITreturns" method="get" >
name:<input type="text" name="name"/>
<select name="gender">
<option>male</option>
<option>fe</option>
```

```
</select>
sal:<input type="text" name="salary"/>
tax:<input type="text" name="tax"/>
<input type="submit"/>
</form>
</body>
</html>
```

8. Write a JSP and Servlet Program to do the following to buy a T-Shirt online:

a. **A set of checkboxes to select your T-Shirt accessories such as ‘belt’, ‘cap’, ‘hair-band’ etc.**
b. **A text area / text field to enter your T-Shirt tag-line**
c. **A Radio-button that allows the user to choose between T-Shirt with chest pocket and without.**
d. **A Combo Box to choose your T-Shirt color**
e. **Appropriate labels for these GUI Components**
f. **A Button called “Click Me” which when pressed will**
g. **Insert the details entered into a table called ‘TShirts’.**
h. **An OrderNo is generated by adding ‘1’ to the existing ‘OrderNo’**
i. **If ‘TShirts’ table is empty the initial value of ‘OrderNo’ is 100.**
j. **This ‘OrderNo’ is also inserted into the ‘TShirts’ table**
k. **Display all the records of the ‘TShirts’ table in tabular form**

PS: Frontend display should be in JSP and the business logic should be written in Servlet Class.

```
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.*;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet("/tcon")
public class tcon extends HttpServlet {
private static final long serialVersionUID = 1L;

public tcon() {
   super();
}
```

```
    protected void service(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
            response.setContentType("text/html");
            PrintWriter out=response.getWriter();
            String[] Accessories={};
            Accessories=request.getParameterValues("access");
            String tshirtAccessories="";
            String tshirtTagLine=request.getParameter("tagline");
            String tshirtOption=request.getParameter("pocket");
            String tcolor=request.getParameter("Tshirtcolor");
            out.println("<html>");
            out.println("<head><title>T-shirt</title></head>");
            out.println("<body>");
            try {
        Statement stmt;
        Class.forName("com.mysql.jdbc.Driver");
                                            Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/tshrit", "root", "");
        if (conn != null) {
            stmt= conn.createStatement();
            String qu;
            if(tshirtAccessories!=null && tshirtTagLine!=null && tshirtOption!=null
&& tcolor!=null){
                for(String option:Accessories){
                    tshirtAccessories=tshirtAccessories+option;
                }
                qu="insert into Tshirts
values("+null+",'"+tshirtTagLine+"','"+tshirtAccessories+"','"+tcolor+"','"+tshirtOption+"');";
                stmt.executeUpdate(qu);
            }
            qu="select * from Tshirts;";
            ResultSet rs =stmt.executeQuery(qu);
            out.println("<table border=2>");
            out.println("<tr>");
            out.print("<td>OrderNo</td>");
            out.print("<td>T-shirt Accessories</td>");
            out.print("<td>T-shirt tag-line</td>");
            out.print("<td>T-shirt type</td>");
            out.print("<td>T-shirt color</td>");
            out.println("</tr>");
            if(!rs.isBeforeFirst()){
                out.print("<tr>");
                        out.print("<td>100</td>");
                        out.print("<td>NULL</td>");
                        out.print("<td>NULL</td>");
                        out.print("<td>NULL</td>");
                        out.print("<td>NULL</td>");
                        out.print("<td>NULL</td>");
                        out.println("</tr>");
            }
            while(rs.next()){
            out.println("<tr>");

out.print("<td>"+(Integer.parseInt(rs.getString("OrderNo"))+100)+"</td>");
                out.print("<td>"+rs.getString("tshritAccessories")+"</td>");
                out.print("<td>"+rs.getString("tshritTagLine")+"</td>");
                out.print("<td>"+rs.getString("tcolor")+"</td>");
                out.print("<td>"+rs.getString("tshritOption")+"</td>");
                out.println("</tr>");
            }
            out.println("</table>");
            out.println("<a href=\"tshrit.jsp\">click here</a>");
            out.println("</body></html>");
        }
            }
            catch (Exception e){
                e.printStackTrace();
            }
        }
    }
    ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    tshrit.jsp
    ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    <html>
    <head>
    <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
    <title>Insert title here</title>
    </head>
    <body>
    <form action="tcon" method="post">
    T-Shirt Accessories<input type="checkbox" name="access"  value="Belt"/>Belt
    <input type="checkbox" name="access" value="Cap"/>Cap
    <input type="checkbox" name="access" value="Hair-Band"/>Hair-Band<br>
    Tag-Line<input type="text" name="tagline" size="50"/><br>
    T-Shirt Feature:<input type="radio" name="pocket" value="ChestPocket"/>Chest Pocket
    <input type="radio" name="pocket" value="NoChestPocket"/>No Chest Pocket<br>
    T-Shirt Color:<select name="Tshirtcolor">
    <option>Blue</option>
    <option>Red</option>
    <option>Green</option>
```

```
</select><br>
<input type="submit" value="Place Orders"/>
</form>
</body>
</html>
```

9. Create a Telephone Directory Application using Servlet that searches the database based on phone number or name. Also show database table creation with inserting 2-3 values to the table.

a. **Database Name: OnlineDirectory**
b. **Table Design:**
   i. **Table Name: Telephone_Directory**
   ii. **Attributes: Phone_Number, Name, Address, Company, Pin_Code.**

JDBClogin.java

```
~~~~~~~~~~~~~~~~~~~~~~~~~
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.Statement;
import java.sql.DriverManager;
import java.sql.SQLException;

import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@SuppressWarnings("serial")
@WebServlet(urlPatterns={"/javaConnection"})
public class JDBClogin extends HttpServlet {

    static Connection getConnection() throws Exception {

        String driver = "com.mysql.jdbc.Driver";
        String url = "jdbc:mysql://localhost/onlinedirectory";
        String username = "root";
        String password = "";
        Class.forName(driver);
        Connection conn = DriverManager.getConnection(url, username, password);
        return conn;
    }

    public void doGet(HttpServletRequest request, HttpServletResponse response) throws IOException {
        PrintWriter out = response.getWriter();
        //out.print("Working");

          boolean flag = false;
        Connection conn = null;
```

```
        Statement stmt = null;
        java.sql.ResultSet rs = null;
    try {
            conn = getConnection();
            stmt = conn.createStatement();
            out.print("Working");
                long inp;

            try
            {
             inp =Long.parseLong(request.getParameter("phone"));
            out.println(""+inp);
             rs = stmt.executeQuery("SELECT * FROM tele_dir where contact="+inp);
            }
            catch(Exception e)
            {
                String name=request.getParameter("phone");
                // out.println(""+name);
                rs = stmt.executeQuery("SELECT * FROM tele_dir where name='"+name+"'");

            }
            if(rs.next()) {
                String name = rs.getString(1);
                long contact = rs.getLong(2);
                String address = rs.getString(3);
                String company = rs.getString(4);
                int pin =rs.getInt(5);
                out.println("name"+name);
                out.println("contact:"+contact);
                out.println("address:"+address);
                out.println("company:"+company);
                out.println("pin:"+pin);


                }
            else
            {
                out.println("no contact found");
            }


        } catch (ClassNotFoundException e) {
            System.out.println("Error: failed to load MySQL driver.");
            e.printStackTrace();
        } catch (SQLException e) {
            System.out.println("Error: failed to create a connection object.");
            e.printStackTrace();
        } catch (Exception e) {
            System.out.println("Error: unknown");
            e.printStackTrace();
        }

    finally {
        try {
            stmt.close();
            conn.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }


    }

}
~~~~~~~~~~~~~~~~~~~~~~~~~
insert1.java
~~~~~~~~~~~~~~~~~~~~~~~~~
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.Statement;
import java.sql.DriverManager;
import java.sql.SQLException;

import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@SuppressWarnings("serial")
@WebServlet(urlPatterns = { "/ins" })
public class insert1 extends HttpServlet {

    static Connection getConn() throws Exception {
        String driver = "com.mysql.jdbc.Driver";
        String url = "jdbc:mysql://localhost/onlinedirectory";
        String username = "root";
        String password = "";
        Class.forName(driver);
        Connection conn = DriverManager.getConnection(url, username, password);
        return conn;
    }

    Connection conn1 = null;

    public void doGet(HttpServletRequest request, HttpServletResponse response)
            throws IOException {
        PrintWriter out = response.getWriter();
        // out.print("Working");
```

```
    boolean flag = false;
    Connection conn = null;
    Statement stmt = null;
    java.sql.ResultSet rs = null;

    try {
      // conn = getConn();
      String driver = "com.mysql.jdbc.Driver";
      String url = "jdbc:mysql://localhost:3306/onlinedirectory";
      String username = "root";
      String password = "";
      Class.forName(driver);
      conn1 = DriverManager.getConnection(url, username,
          password);
      if (conn1 != null)
        System.out.println("Successful");
      stmt = conn1.createStatement();

      out.print("Working");

      String name = request.getParameter("nam");
      long contact = Long.parseLong(request.getParameter("cnt"));
      String address = request.getParameter("address");
      String company = request.getParameter("company");
      int pin = Integer.parseInt(request.getParameter("pin"));

      out.println("name" + name);
      out.println("contact:" + contact);
      out.println("address:" + address);
      out.println("company:" + company);
      out.println("pin:" + pin);

      stmt.executeUpdate("insert into tele_dir values('" + name + "'," + contact + ",'" + address + "','" +
company + "'," + pin + ");");
      out.println("updated the records");

    } catch (ClassNotFoundException e) {
      System.out.println("Error: failed to load MySQL driver.");
      e.printStackTrace();
    } catch (SQLException e) {
      System.out.println("Error: failed to create a connection object.");
      e.printStackTrace();
    } catch (Exception e) {
      System.out.println("Error: unknown");
      e.printStackTrace();
    } finally {
      try {
        stmt.close();
        conn1.close();
```

```
            } catch (Exception e) {
                e.printStackTrace();
            }
        }

    }

}
~~~~~~~~~~~~~~~~~~~~~~~~~
Index.jsp
~~~~~~~~~~~~~~~~~~~~~~~~~
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<form  action="javaConnection" method="get"/>
Enter name or phone:<input type="text" name="phone" /><br/>
<input type="submit" />
</form>
<a href="insert.html;">insert into directory</a>
</body>
</html>
~~~~~~~~~~~~~~~~~~~~~~~~~
insert.html
~~~~~~~~~~~~~~~~~~~~~~~~~
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<form action="ins" method="get" >
name:<input type="text" name="nam" /><br/>
contact:<input type="text" name="cnt" /><br/>
address:<input type="text" name="address" /><br/>
company:<input type="text" name="company" /><br/>
pincode:<input type="text" name="pin" />
<input type="submit" />
</form>
</body>
</html>
```

10. Write a Java Program that creates two threads object of Thread class. Where one thread asks the user to enter a number not less than four digits. Split the digits of the number and display in words the value of the number. Ex: 1 – One. Second thread finding the number of vowels in a word. Ex: JAVA – Vowel - A, Count – 2.

GetStringThread.java

```
~~~~~~~~~~~~~~~~~~~~~~~~~~
import java.util.Scanner;

public class GetStringThread extends Thread {

public String string;
public static String vowels = "aeiou";
public void run() {
Scanner s = new Scanner(System.in);
System.out.println("Enter a string: ");
string = s.next();
int x = 0;
for(char c : string.toCharArray())
for(char ch : vowels.toCharArray())
if(c==ch) x++;
System.out.print(x+" vowels present\n");
}
}
~~~~~~~~~~~~~~~~~~~~~~~~~~
GetNumberThread.java
~~~~~~~~~~~~~~~~~~~~~~~~~~
import java.util.Scanner;

public class GetNumberThread extends Thread {

public String number;
private static String[] digit = new String[]
{"zero","one","two","three","four","five","six","seven","eight","nine"};
public void run() {
Scanner s = new Scanner(System.in);
System.out.println("Enter number with more than 4 digits: ");
number = s.next();
for(char c : number.toCharArray()) {
if(c<48||c>57) {
System.out.println("Invalid inputs");
break;
}
System.out.print(digit[((int)c-48)]+" ");
}
System.out.print("\n");
}
}
~~~~~~~~~~~~~~~~~~~~~~~~~~
InputThreads.java
~~~~~~~~~~~~~~~~~~~~~~~~~~
public class InputThreads {

        public static void main(String[] args) throws InterruptedException {
        GetStringThread getStringThread;
```

```
        GetNumberThread getNumberThread;
        getNumberThread = new GetNumberThread();
        getStringThread = new GetStringThread();
        //getNumberThread.start();
        getStringThread.start();
        Thread.sleep(100);
        //getStringThread.start();
        getNumberThread.start();
        }
}
```

11. Write a program using JSP that helps a student to calculate the income tax on various annual incomes that he will be earning when he gets a job.

Login.html **will call** ***dataCapture.jsp*** **that should do the following:**

- **Use Java Collections to make a list of valid users and facilitate user login functionality.**
- **Give a personalized Welcome message and display today's date.**
- **Have a Text Entry with label 'Name' to enter the name of the user.**
- **Give a List of Organizations to choose 'Place of Work'**
- **Provide a Male or Female option to choose the 'Gender'**
- **Have a Text Entry with label 'Annual Income'**
- **Give a Submit button reading 'Calculate Tax'**

CalculateTax.jsp **must calculate the interest based on the following business rules:**

- **Salary below 1,00,000 shall no have income-tax.**
- **Calculate 15% of tax on 1,00,001 – 5,00,000.**
- **Calculate 20% on 5,00,001 onwards.**

The final income tax along with the details of how it is calculated must be put in a session object and displayed to the user in ***dataCapture.jsp*****. All the income taxes calculated so far by the user, must be taken out of the session object and displayed, each time in** ***dataCapture.jsp*** **which has a link called '*****Logout*****' that destroys the session.**

Login.jsp:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Login Page</title>
</head>
<body>
```

```
    <form action="dataCapture.jsp" method="post">
        User Name: <input type="text" size="15" name="username"> <br>
        Password: <input type="password" size="15" name="password"> <br>
        <input type="submit" value="Login">
    </form>
    <%
        String reason = request.getParameter("FailReason");
        if (reason != null)
            out.println(reason);
    %>
</body>
</html>
```

dataCapture.jsp:

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1" import="java.util.* , java.text.*"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
</head>
<body>

    <%!HashMap hm;
    String uname;
    String pwd;
    Map.Entry entry;
        boolean login = false;
        hm = new HashMap();
        uname = request.getParameter("username");
        pwd = request.getParameter("password");

        hm.put("Archie", "Riverdale");
        hm.put("Haddock", "Marlinspike");
        hm.put("Hermione", "Hogwarts");

        Set s = hm.entrySet();
        Iterator it = s.iterator();

        while (it.hasNext())
        {
            entry = (Map.Entry) it.next();
            if (uname.equals(entry.getKey())
                            && pwd.equals(entry.getValue()))
            {
                    login = true;
            }
        }//end while

        if (login == true)
        {
            out.println("<B><FONT COLOR = Blue>");
            out.println("Welcome </FONT></B>");
            out.println(uname);
            DateFormat dateFormat = new SimpleDateFormat(
                            "yyyy/MM/dd HH:mm:ss");
            Date date = new Date();
            out.println("<BR><FONT COLOR = Green>");
            out.println("Today is </FONT>" + dateFormat.format(date));
    %>

    <form action="CalculateInterest.jsp" method="post">
        <FONT COLOR="Magenta"> First Name:</FONT> <input type="text"
size="15"
            name="fname"> <br> <FONT COLOR="Brown">Last Name:
        </FONT> <input type="text" size="15" name="lname"> <br> <FONT
            COLOR="Purple">Select your Place of Work:</FONT> <br> <select
            name="profession" size="3">
            <option>IT Company</option>
            <option>Private Bank</option>
            <option>Insurance Company</option>
        </select> <br> <input type="radio" name="gender" value="Male">
        Male<br> <input type="radio" name="gender" value="Female"
checked>Female<br>
        <br> <FONT COLOR="Red"> Annual Income(in Rupees):</FONT> <input
            type="text" size="15" name="income"> <br> <br> <input
            type="submit" value="Calculate Tax">
    </form>
    <%
        }
        else
        {
    %>
    <jsp:forward page="Login.jsp">
        <jsp:param name="FailReason" value="Wrong Username or Password" />
    </jsp:forward>
    <%
        }
    %>

</body>
</html>
```

CalculateInterest.jsp:

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Calculate Interest JSP</title>
</head>
<body>
    <%
            String fname = request.getParameter("fname");
            String lname = request.getParameter("lname");
            String gender = request.getParameter("gender");
            String profession = request.getParameter("profession");
            String prefix = " ";

            if (gender.equals("Male"))
            {
                    prefix = "Mr.";
            } else if (gender.equals("Female"))
            {
                    prefix = "Ms.";
            }
    %>
    <FONT COLOR="Blue">Hello
<%=prefix%>&nbsp;<%=fname%>&nbsp;<%=lname%>
            &nbsp; who works in a <%=profession%></FONT>

    <%
            String sincome = request.getParameter("income");
            float income = Float.parseFloat(sincome);
            out.println("<BR>Your Annual Income is " + income);

            float tax;
            float diff;

            if (income <= 100000)
            {
```

```
          out.println("<BR>You are below the Tax Bracket!!");
      }
      else if (income > 100000 && income <= 200000)
      {
          out.println("Your Tax Bracket is between Rs.1,00000 to Rs.2,00000");
          out.println("<BR>Tax to be paid is 10% of income above 1Lakh");
          diff = income - 100000;
          tax = (float) 0.1 * diff;
          out.println("<BR>Tax to be paid is " + tax);
      }
      else if (income > 200000 && income <= 300000)
      {
      out.println("<BR>Your Tax Bracket is between between Rs.1,00000 to
Rs.3,00000");
  out.println("<BR>Tax to be paid is 10% of income upto 1Lakh and 20% of rest of
income");
          diff = income - 200000;
          tax = (float) 0.2 * diff + (float) 0.1 * 100000;
          out.println("<BR>Tax to be paid is " + tax);
      }
      else if (income > 100000 && income <= 400000)
      {
          out.println("<BR>Your Tax Bracket is between Rs.1,00000 to
Rs.4,00000");
          out.println("<BR>Tax to be paid is 10% of income upto 1Lakh 20% of
income upto 3Lakh and 30% of rest of income");
          diff = income - 300000;
          tax = (float) 0.3 * diff + (float) 0.2 * 200000 + (float) 0.1
                  * 100000;
          out.println("<BR>Tax to be paid is " + tax);
      }
      else if (income > 400000)
      {
          out.println("<BR>You fall in the tax bracket greater than Rs.4,00000");
          diff = income - 400000;
          tax = diff + (float) 0.3 * 300000 + (float) 0.2 * 200000
                  + (float) 0.1 * 100000;
          out.println("<BR>Tax to be paid is 10% of income upto 1Lakh 20% of
income upto 3Lakh, 30% of income upto 4 lakh and 100% of rest of income");
          out.println("<BR>Tax to be paid is " + tax);
      }//end if
  %>
</body>
</html>
```

12. a. Create two tables Flight(Flight_Number, Airline_Name, Weekdays) and SeatReservation(Flight_Number, Date, Seat_Number, Customer_Name, Customer_Phone) in MySQL database.
b. Create JSP page *ReserveOnline.jsp* to reserve an airline seat and insert the values into the table SeatReservation. OnClick of Submit in *ViewDetails.jsp* display information about reservation. Validate the Flight_Number from already existing Flight database and generate random number for Seat_Number within the range 1-500.
c. Also create a link to display information of all the flights running on a particular day.

```
flightdete.jsp
~~~~~~~~~~~~~~~~~~~
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
<body>
        <form action="flidet.jsp" method="post">
                <p>Enter the day for flight details</p>
                <input type="text" name="fdate" size="10" /> <input type="submit"
                        value="Click" />
        </form>
</body>
</html>
~~~~~~~~~~~~~~~~~~~
flidet.jsp
~~~~~~~~~~~~~~~~~~~
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
<body>
        <%
                String fldate = request.getParameter("fdate");
                String url = "jdbc:mysql://localhost/flight";
                String user = "root";
                String password = "";
                Connection connection = null;
                Statement stmt;
                try
                {
```

```
            Class.forName("com.mysql.jdbc.Driver").newInstance();
            out.println("hi, Flight details=");
            connection = DriverManager.getConnection(url, user, password);
            if (connection != null)
            {
                stmt = connection.createStatement();
                String query = " select * from Flight where Weekdays='"
                        + fldate + "'";
                ResultSet re = stmt.executeQuery(query);
                while (re.next())
                {
                    out.println(re.getString("Flight_Number") + "\n"
                            + re.getString("Airline_Name") + "\n"
                            + re.getString("Weekdays") + "\n");
                }
            }
            else
                out.println("Connection refused");
        }
        catch (Exception e)
        {
            out.println(e.getMessage());
        }
    %>
</body>
</html>
~~~~~~~~~~~~~~~~~~~~
ReserveOnline.jsp
~~~~~~~~~~~~~~~~~~~~
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
    <%
        out.println("<B><FONT COLOR = Blue>");
        out.println("Welcome </FONT></B>");
        DateFormat dateFormat = new SimpleDateFormat("yyyy/MM/dd HH:mm:ss");
        //Date datess = new Date();
        out.println("<BR><FONT COLOR = Green>");
        //out.println("Today is </FONT>"+dateFormat.format(datess));
    %>
```

```
    <form action="ViewDetails.jsp" method="post">
        <FONT COLOR="Magenta"> Flight Number:</FONT> <input type="text"
            size="15" name="fname"> <br> <FONT COLOR="Brown">Date:
        </FONT> <input type="text" size="15" name="date"> <br> <FONT
            COLOR="Brown">Customer Name: </FONT> <input type="text"
size="15"
            name="custname"> <br> <FONT COLOR="Brown">Customer
            Number: </FONT> <input type="text" size="15" name="custno"> <br>
        <input type="submit" value="Submit form">
    </form>
</body>
</html>
~~~~~~~~~~~~~~~~~~~
ViewDetails.jsp
~~~~~~~~~~~~~~~~~~~
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
<body>
    <%
        String fno1 = request.getParameter("fname");
        int fno = Integer.parseInt(fno1);
        String datea = request.getParameter("date");
        Random rand = new Random();
        int s = rand.nextInt(500);
        String seat = Integer.toString(s);
        String custname = request.getParameter("custname");
        String custno = request.getParameter("custno");
        int fno2 = Integer.parseInt(custno);
    %>

    <%
        String url = "jdbc:mysql://localhost/flight";
        String user = "root";
        String password = "";
        Connection connection = null;
        //boolean flag=false;
        String query2;
        boolean h = false;
        Statement stmt;
        try
```

```
                {
                        Class.forName("com.mysql.jdbc.Driver").newInstance();
                        connection = DriverManager.getConnection(url, user, password);
                        if (connection != null)
                        {
                                out.println("Connection created");
                                stmt = connection.createStatement();
                                query2 = "select * from Flight where Flight_Number='" + fno
                                                + "'";
                                ResultSet rs = stmt.executeQuery(query2);
                                while (rs.next())
                                {
                                        h = true;
                                }
                                if (h)
                                {
                                        out.println("Flight Number:" + fno + "\nDate:" + datea
                                                        + "\nSeatNumber:" + seat +
"\nCustomerName:"
                                                        + custname + "\nCustNumber:" + custno);
                                        String query = "insert into SeatReservation values('"
                                                        + fno + "','" + datea + "','" + seat + "','"
                                                        + custname + "','" + custno + "');";
                                        stmt.executeUpdate(query);
                                        out.println("\nDetails inserted");
                                }
                                else
                                {
                                        out.println("flight number doesnot exist");
                                }
                        }
                        else
                                out.println("Connection refused");
                }
                catch (Exception e)
                {
                        out.println(e.getMessage());
                }
        %>
        <p>
                click for flight details <a href="flightdete.jsp">here</a>
</body>
</html>
```

Text books

1. Herbert Schildt, ‘The Complete Reference Java (J2SE 5 Edition)’, TATA McGRAW-HILL Edition 2005.
2. Ivan Bayross, Sharanam Shah, Cyntiha Bayross and Vishali Shah, ‘Java EE 5 for Beginners’, SPD (Sharoff Publishers & Distributors Pvt. Ltd.), 2nd edition August 2008.

Course Delivery:

The Course will be delivered through practice exercises, presentation

Course Assessment and Evaluation:

	What		To whom	When/ Where (Frequency in the course)	Max marks	Evidence collected	Contributing to Course Outcomes
Direct Assessment Methods	**CIE**	Internal assessment tests	Students	Twice(sum of two will be computed)	30	Data Sheets	C01
		Miniproject- Demo + Viva		Once	10	Presentation	CO2
		Documentation – Report + Record		Once	10	Report + Record	CO3
	SEE	Standard examination		End of course	50	Answer scripts	CO1, CO3
Indirect Assessment Methods	End of course survey		Students	End of course	-	Questionnaire	C01, C02, C03

COURSE OUTCOMES

The students will be able to

CO1: Design and Develop Java programs using the Java concepts such as packages, inheritance, interfaces, exception handling, collection frameworks, Swings, Servlets, JSP, JDBC and Multithreading. **(PO- b,d)**

CO2: Apply java concepts in various domains. **(PO-d)**

CO3. Produce a substantial written documentation. **(PO-j)**

Rubrics for Miniproject Evaluation