|  |
| --- |
| **Java and J2EE Lab** |
| Course Code: ISL67  Credit: 0:0:1 Contact Hours: 28 |

**Course Content**

|  |  |
| --- | --- |
| 1. 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 caller 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.   Numbers are recalled in the order they arrive  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.  After each number display, the user can select   * + 1. To delete the call     2. To go on to the next missed call, or     3. To display the call details (number, caller name and time).   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.\*;  **class** CallerD  {  Date date = **new** Date();  String name;  Integer telNo;  CallerD(Date d, String n,Integer t)  {  date =d;  name = n;  telNo = t;  }  }  **public** **class** TelephoneClass {  /\*\*  \* **@param** args  \*/  HashMap<Long,String> hm = **new** HashMap<Long,String>();  ArrayList<Long> callList = **new** ArrayList<Long>();  ArrayList<CallerD> missedcall = **new** ArrayList<CallerD>();  **void** createCaller()  {  hm.put(**new** Long("9898080099"),"Hari M");  hm.put(**new** Long("9898087654"),"Lankesh M");  hm.put(**new** Long("9898897843"),"Lucky");  }    **public** **static** **void** main(String[] args) {  // **TODO** Auto-generated method stub  }  }  //Program to create a database of callers  import java.util.\*;  import java.util.Map.Entry;  public class DataBaseCallers {  /\*\*  \* @param args  \*/  public static HashMap<Long,String> hm;  public static Scanner sc = new Scanner(System.in);  static void createEnteries(int n)  { hm = new HashMap<Long,String>();  for(int i=0;i<n;i++)  {  System.out.println("Enter the phone number and Contact name");  Long phno = sc.nextLong();  String name = sc.next();  hm.put(phno, name);    }  }  public static void ForHoldingData() {  // TODO Auto-generated method stub  System.out.println("Enter how many details do you require?");  int n = sc.nextInt();  createEnteries(n);  }  public static void display()  {  Set<Entry<Long,String>> hashSet=hm.entrySet();  for(Entry<Long,String> entry:hashSet ) {  System.out.println("Key="+entry.getKey()+", Value="+entry.getValue());  }  }  }  **import** java.util.\*;  **class** MissedCallDetails  {  Calendar calObj;  Long tel\_num;  String name;  MissedCallDetails(Calendar cob, Long tn, String n)  {  calObj = cob;  tel\_num = tn;  name = n;  }  **void** display()  { calObj.add(Calendar.*DATE*, 1);  //SimpleDateFormat format1 = new SimpleDateFormat("yyyy-MM-dd");  System.*out*.println("Current Time is "+calObj.getTime());  System.*out*.println("Telephone number is "+tel\_num);  System.*out*.println("Name is "+name);  }  }  **public** **class** RecordManageMissedCall {  /\*\*  \* **@param** args  \*/  **static** LinkedList<MissedCallDetails> *amiss*;  **static** **void** HandleMissedCallActivities()  {  *amiss* = **new** LinkedList<MissedCallDetails>();  **while**(**true**)  {  System.*out*.println("Enter a choice");  System.*out*.println("1: add missed call \n 2.Display and delete on request \n 3. Delete based on number \n 4. Print");  **int** choice = DataBaseCallers.*sc*.nextInt();  //int i=1;  **switch**(choice)  {  **case** 1: //add missed call  System.*out*.println("Enter the missed call telephone number");  Calendar cb = Calendar.*getInstance*();  Long telnum= DataBaseCallers.*sc*.nextLong();  String name; // = DataBaseCallers.sc.next();  //check for name in the Database  **if**(DataBaseCallers.*hm*.containsKey(telnum))  name = DataBaseCallers.*hm*.get(telnum);  **else**  name = "Private Caller";  MissedCallDetails mcd = **new** MissedCallDetails(cb,telnum,name);  **if**(*amiss*.size()==10)//for eleventh entry onwards  {  //for eleventh entry onwards enter from the beginning  *amiss*.removeLast();    }    *amiss*.addFirst(mcd);    **break**;  **case** 2: // Display number and ask for user to delete  ListIterator<MissedCallDetails> it = *amiss*.listIterator();  LinkedList<MissedCallDetails> removeList = **new** LinkedList<MissedCallDetails>();  **int** i=0;  **while**(it.hasNext())  { i++;    System.*out*.println("Number is ");  MissedCallDetails m1 = it.next();  System.*out*.println(m1.tel\_num);  System.*out*.println("Do you want to delete the details related to this number? Indicate by 1 : delete, 2: move next call , 3: display call details \n");  **int** cho = DataBaseCallers.*sc*.nextInt();  **if**(cho==1)  removeList.add(m1);  **else** **if**(cho==3)  {//display  m1.display();  }  }  **if**(i==0)  System.*out*.println("No missed calls");  *amiss*.removeAll(removeList);    **break**;  **case** 3: //Delete based on the number specified by the user  System.*out*.println("Delete based on the number given by user");  System.*out*.println("Enter the number");  Long num = DataBaseCallers.*sc*.nextLong();  ListIterator<MissedCallDetails> it1 = *amiss*.listIterator();  **boolean** flag = **false**;  i=0;  **while**(it1.hasNext())  { i++;  MissedCallDetails m1 = it1.next();  **if**(m1.tel\_num==num)  {  flag =**true**;  *amiss*.remove(m1);  **break**;  }  }  **if**(i!=0)  {  **if**(flag==**true**)  System.*out*.println("Phone number with details "+ num +"deleted");  **else**  System.*out*.println("No such number exists");  }  **else**  System.*out*.println("No missed Call");  **break**;  **case** 4:  //print missed call details  ListIterator<MissedCallDetails> it2 = *amiss*.listIterator();  **while**(it2.hasNext())  {  MissedCallDetails m1 = it2.next();  m1.display();  }  **break**;    **default**: **return**;  }  }  }    **public** **static** **void** main(String[] args) {  // **TODO** Auto-generated method stub  //1. Enter contact details in the database  DataBaseCallers.*ForHoldingData*();  System.*out*.println("Receive missed Calls");  //DataBaseCallers.sc  *HandleMissedCallActivities*();  DataBaseCallers.*display*();    }  } | |
| 1. Write a Java program using user-defined storage classes to create a book database and store it in a Collection List. Books collection should include title, author, publisher and price. Also write a method to sort the books in ascending order of price and store it in another List. Maintain the book details with respect to an unique book id. Prompt for an author name and list all the books with the same author name. Create a new list holding all the book details with price greater than a user specified price. For a given a value by the user, find all the books that match either the whole or a part of the book title. Identify a publisher and print books from a particular publisher. Update the publisher details based on a title.   import java.util.\*;  class Book  {  String title,author, publisher;  double price;  Book(String title,String author,String publisher,double price)  { this.title=title;  this.author=author;  this.publisher=publisher;  this.price = price;  }  public String toString()  {  String str= "Book has "+title+"title "+"whose author is "+author;  str+="\n Book is published by "+publisher;  str+="\n Price is= "+price;  return str;  }    }  class BookCompare implements Comparator<Book>  {  public int compare(Book b1,Book b2)  {  return (int) (b1.price-b2.price);  }  }  public class SecondProg\_lab {  static HashMap<Integer,Book> hmbook = new HashMap<Integer,Book>();  static Scanner sc = new Scanner(System.in);  void BookHashmap(ArrayList<Book> barr)  { int i=1;  for(Book b1:barr)  {  hmbook.put(i,b1);  i++;  }      }  static void printHashMap()  {  //print hashmap  Set<Map.Entry<Integer,Book>> set = hmbook.entrySet();  for(Map.Entry<Integer, Book> s:set)  {  System.out.println(s.getKey());  System.out.println(s.getValue());  }    }    static void printBookData(Book b)  {  String str= "Book has "+b.title+"title "+"whose author is "+b.author;  str+="\n Book is published by "+b.publisher;  str+="\n Price is= "+b.price;  System.out.println(str);  }    static void Read\_data(ArrayList<Book> barr)  {    while(true)  {  System.out.println("Do u want to enter book details? Mention yes/no");  String val = sc.nextLine();  if(val.toUpperCase().equals("YES"))  {  System.out.println("Enter Book Details");  System.out.println("Enter Title");  String title =sc.nextLine();  System.out.println("Enter Author");  String author = sc.nextLine();  System.out.println("Enter Publisher");  String pub = sc.nextLine();  System.out.println("Enter Price");  double price = Double.parseDouble(sc.nextLine());  Book b1 = new Book(title,author,pub,price);  barr.add(b1);    }  else  {  break;  }  }    }  static boolean isSubString(String s1,String s2)  {  int M = s1.length();  int N = s2.length();    /\* A loop to slide pat[] one by one \*/  for (int i = 0; i <= N - M; i++) {  int j;    /\* For current index i, check for  pattern match \*/  for (j = 0; j < M; j++)  if (s2.charAt(i + j) != s1.charAt(j))  break;    if (j == M)  return false;  }    return true;  }    public static void main(String args[]){    ArrayList<Book> arbook = new ArrayList<Book>();  Read\_data(arbook);  //Sort in ascending order  ArrayList<Book> clone = (ArrayList<Book>) arbook.clone();  ArrayList<Book> booklist = clone;  booklist.sort(new BookCompare());  System.out.println("Before sorting");  System.out.println(arbook);  System.out.println("After sorting");  System.out.println(booklist);  //Prompt for an author name and list all the books with the same author name.  System.out.println("Enter author name");  String aut\_name = sc.nextLine();  Set<Map.Entry<Integer,Book>> set = hmbook.entrySet();  for(Map.Entry<Integer, Book> s:set)  { Book b1=s.getValue();  String s1 =b1.author;  if(s1.equals(aut\_name))  {  printBookData(s.getValue());    }    }  //Create a new list holding all the book details with price greater than a user specified price.  LinkedList<Book> lbook = new LinkedList<Book>();  System.out.println("To list all books with price greater than a value, enter a price");  double pri = Double.parseDouble(sc.nextLine());  set = hmbook.entrySet();  for(Map.Entry<Integer, Book> s:set)  { Book b1=s.getValue();  if(b1.price>pri)  {  lbook.add(b1);    }    }  //print linkedlist  System.out.println("Books with price > than "+pri);  for(Book b:lbook)  {  System.out.println(b);  }  //For a given a value by the user, find all the books that match either the whole or a part of the book title.  System.out.println("enter a part of a publishers name");  String p = sc.nextLine();  set = hmbook.entrySet();  for(Map.Entry<Integer, Book> s:set)  { Book b1=s.getValue();  if(isSubString(b1.title,p))  {  printBookData(b1);    }    }  //Identify a publisher and print books from a particular publisher.  System.out.println("enter a publishers name to print book details");  p = sc.nextLine();  set = hmbook.entrySet();  for(Map.Entry<Integer, Book> s:set)  { Book b1=s.getValue();  if(b1.publisher.equals(p))  {  printBookData(b1);    }    }  //Update the publisher details based on a title.  System.out.println("enter a title whose publisher is to be updated");  String t = sc.nextLine();  System.out.println("enter the updated publishers name");  p = sc.nextLine();  set = hmbook.entrySet();  for(Map.Entry<Integer, Book> s:set)  { Book b1=s.getValue();  if(b1.title.equals(t))  {  b1.publisher=p;  hmbook.put(s.getKey(), b1);  //printBookData(b1);  }    }  printHashMap();  }    } | |
| 1. Create a desktop java application using swings to enable an user to enter student information such as name, usn, age, address, sgpa of 8 semesters, category. Perform validations on age and sgpa. Display appropriate messages in pop up boxes to indicate wrong entries, on clicking of the “compute” button. Also find the cgpa based on the obtained sgpa. On clicking of the “done” button, place the student details in a collection. A click on the “display” button should display the collection in a textarea.   **public** **class** Student {  String name,usn,address,category;  **int** age;  //float sgpa1,sgpa2,sgpa3,sgpa4,sgpa5,sgpa6,sgpa7,sgpa8;  **float** cgpa;  **public** Student(String name,String usn,String address,String cat,**int** age, **float** cgpa)  {  **this**.name=name; **this**.usn=usn;  **this**.address=address; **this**.category=cat;  **this**.age=age;  **this**.cgpa=cgpa;  }  **public** String toString()  {  String stud= name + " "+usn+" residing in "+address+" belonging to category "+category+" of age "+age;  stud+= "has cgpa "+cgpa;  **return** stud;  }  }  import java.awt.\*;  import java.awt.event.\*;  //import java.awt.event.ActionListener;  import java.util.LinkedList;  import javax.swing.\*;  public class StudentClass implements ActionListener{  LinkedList<Student> stud\_list = new LinkedList<Student>();  JLabel jl1 = new JLabel("Enter Name");  JLabel jl13 = new JLabel("Enter usn");  JLabel jl2 = new JLabel("Enter Age");  JLabel jl3 = new JLabel("Enter Address");  JLabel jl4 = new JLabel("Select category");  JLabel jl5 = new JLabel("Enter sgpa of I sem");  JLabel jl6 = new JLabel("Enter sgpa of II sem");  JLabel jl7 = new JLabel("Enter sgpa of III sem");  JLabel jl8 = new JLabel("Enter sgpa of IV sem");  JLabel jl9 = new JLabel("Enter sgpa of V sem");  JLabel jl0 = new JLabel("Enter sgpa of VI sem");  JLabel jl11 = new JLabel("Enter sgpa of VII sem");  JLabel jl12 = new JLabel("Enter sgpa of VIII sem");  JLabel jcgpa = new JLabel("CGPA obtained");  JTextField name=new JTextField(20) ;  JTextField usn=new JTextField(20) ;  JTextField age=new JTextField(3) ;  JTextArea address=new JTextArea(3,4) ;  JComboBox cat=new JComboBox() ;  JTextField cgpa =new JTextField(10);  JTextField sgpa1 = new JTextField(5);  JTextField sgpa2 = new JTextField(5);  JTextField sgpa3 = new JTextField(5);  JTextField sgpa4 = new JTextField(5);  JTextField sgpa5 = new JTextField(5);  JTextField sgpa6 = new JTextField(5);  JTextField sgpa7 = new JTextField(5);  JTextField sgpa8 = new JTextField(5);  JButton submit = new JButton("compute");  JButton done = new JButton("done");  JButton display = new JButton("display");  JTextArea stud\_list\_display = new JTextArea(20,20);  JFrame f1=new JFrame("Student Information");  JFrame f2=new JFrame("Student Collection Display");  StudentClass()  {  //JFrame f1=new JFrame("Student Information");  jl1.setBounds(10,10,10,10);  cat.addItem("GM");  cat.addItem("SC/ST");  cat.addItem("Cat1");  cat.addItem("Cat2");  f1.add(jl1);f1.add(name);  f1.add(jl13);f1.add(usn);  f1.add(jl2);f1.add(age);f1.add(jl3);f1.add(address);  f1.add(jl4);f1.add(cat);  f1.add(jl5);f1.add(sgpa1);f1.add(jl6);f1.add(sgpa2);f1.add(jl7);f1.add(sgpa3);f1.add(jl8);f1.add(sgpa4);  f1.add(jl9);f1.add(sgpa5);f1.add(jl0);f1.add(sgpa6);f1.add(jl11);f1.add(sgpa7);f1.add(jl12);f1.add(sgpa8);  f1.add(jcgpa);f1.add(cgpa);  f1.add(submit);f1.add(done);f1.add(display);  f2.add(stud\_list\_display);  f1.setSize(900,800);  f1.setLayout(new GridLayout(8,8));  f1.setVisible(true);  submit.addActionListener(this);  done.addActionListener(this);  display.addActionListener(this);  }  public void actionPerformed(ActionEvent evt)  {  if(evt.getSource()==submit)  {  //check for validations  try  {  int v1=Integer.parseInt(age.getText());  if(v1<18 || v1>35)  {  String age1=JOptionPane.showInputDialog(null,"Enter valid Age");  age.setText(age1);    }    }  catch(NumberFormatException e) {  JOptionPane.showMessageDialog(f1, "Invalid entry");  age.requestFocus();  }    checkSGPA\_valid(1,sgpa1);  checkSGPA\_valid(2,sgpa2);  checkSGPA\_valid(3,sgpa3);  checkSGPA\_valid(4,sgpa4);  checkSGPA\_valid(5,sgpa5);  checkSGPA\_valid(6,sgpa6);  checkSGPA\_valid(7,sgpa7);  checkSGPA\_valid(8,sgpa8);  float cal\_cgpa = calculate\_cgpa();  cgpa.setText(Float.toString(cal\_cgpa));    }  else if(evt.getSource()==done)//to submit into collection  {  Student s1 = new Student(name.getText(),usn.getText(),address.getText(),String.valueOf(cat.getSelectedItem()),Integer.parseInt(age.getText()),Float.parseFloat(cgpa.getText()));  stud\_list.add(s1);  }  else {  f1.setVisible(false);  f2.setVisible(true);  f2.setSize(500,500);  stud\_list\_display.setText(" ");  for(Student s:stud\_list)  {  stud\_list\_display.append(s.toString()+ "\n");  }  }    }  float calculate\_cgpa()  {  float v1 = Float.parseFloat(sgpa1.getText());  float v2 = Float.parseFloat(sgpa2.getText());  float v3 = Float.parseFloat(sgpa3.getText());  float v4 = Float.parseFloat(sgpa4.getText());  float v5 = Float.parseFloat(sgpa5.getText());  float v6 = Float.parseFloat(sgpa6.getText());  float v7 = Float.parseFloat(sgpa7.getText());  float v8 = Float.parseFloat(sgpa8.getText());  return (v1+v2+v3+v4+v5+v6+v7+v8)/8;  }    void checkSGPA\_valid(int sem,JTextField sgpa)  {  try {    if(Float.parseFloat(sgpa.getText())>10)  {  String v1 = JOptionPane.showInputDialog(null,"Enter an SGPA less than or equal to 10 for sem "+sem);  sgpa.setText(v1);  }  }  catch(NumberFormatException e) {  String v2=JOptionPane.showInputDialog(null, "Please enter SGPA for semester "+sem);  sgpa.setText(v2);  //sgpa.requestFocus();    }  }  public static void main(String args[])  {  StudentClass sc = new StudentClass();      }  } | |
| 1. Write a java program using swings to validate user login information using dialog boxes. Once validated, allow the user to enter the customer id, if the person is a new customer, else check whether the customer exists in a collection and obtain the customer id. The customer id can be obtained given a mobile number. Allow the user to enter the item purchased by giving the item id and quantity purchased. On clicking of a button, the item name and the total cost should appear in the corresponding GUI components. Using option dialog box, indicate the types of discount available for the customer. On clicking on the print button, print the details in information dialog box.   **import** java.awt.event.\*;  **import** java.util.ArrayList;  //import java.awt.event.ActionListener;  **import** java.util.HashMap;  **import** java.util.Scanner;  **import** javax.swing.\*;  **public** **class** UserLogin {  JLabel jl1 = **new** JLabel("Enter customer id");  JTextField cid = **new** JTextField(10);  JLabel jl2 = **new** JLabel("Enter mobile number");  JTextField phone = **new** JTextField(12);  JButton check = **new** JButton("Check phone");  JLabel jl3 = **new** JLabel("Enter item id");  JTextField itemid = **new** JTextField(10);  JLabel jl4 = **new** JLabel("Enter quantity bought");  JTextField quan = **new** JTextField(4);  HashMap<Long,Integer> hm = **new** HashMap<Long,Integer>();  **static** ArrayList<ItemDetails> *ar\_items* = **new** ArrayList<ItemDetails>();  JLabel jl5 = **new** JLabel("Item name");  JTextField itemname= **new** JTextField(30);  JLabel jl6 = **new** JLabel("Item cost");  JTextField itemc= **new** JTextField(4);  JButton display\_item = **new** JButton("Item Details");    **void** add\_collection()  {  hm.put(9887667887L,123);  hm.put(7665436788L,232);  hm.put(647485869L,456);  }  **int** check\_collection(Long val)  {  //Add items into collection  add\_collection();  **if**(hm.containsKey(val)) {  **return** hm.get(val);  }  **else**  **return** -1;    }    UserLogin(){  JFrame f1 = **new** JFrame("Customer information");  JFrame f2 = **new** JFrame("Customer Information");  String uname = JOptionPane.*showInputDialog*(**null**,"Enter username");  String passwd = JOptionPane.*showInputDialog*(**null**,"Enter Password");  **if**(uname.equals("uname@gmail.com")&&passwd.equals("pass"))  {  f1.add(jl2);f1.add(phone); f1.add(check);  f1.setSize(200,200);  f1.setVisible(**true**);  check.addActionListener(**new** ActionListener() {    **public** **void** actionPerformed(ActionEvent evt) {  f1.setVisible(**false**);  f2.add(jl1);f2.add(cid);  f2.add(jl3);f2.add(itemid);  f2.add(jl4);f2.add(quan);    //check collection  **int** v=check\_collection(Long.*parseLong*(phone.getText()));  cid.setText(Integer.*toString*(v));  **if**(v==-1)  {  String cid1 = JOptionPane.*showInputDialog*(**null**,"Enter customer id");  cid.setText(cid1);  }  f2.setVisible(**true**);    }  });    }  **else**  {  JOptionPane.*showMessageDialog*(**null**,"Invalid details Please run the code once again ");  }  display\_item.addActionListener(**new** ActionListener() {  **public** **void** actionPerformed(ActionEvent evt) {  **int** v1 = Integer.*parseInt*(itemid.getText());  check\_item(v1);  }  });  }  **void** check\_item(**int** id)  {  **for**(ItemDetails item:*ar\_items*)  {  **if**(item.item\_id==id)  {  itemname.setText(item.item\_name);  itemc.setText(Float.*toString*(item.cost));  }    }  }  **public** **static** **void** main(String args[])  {  //create a collection of items  Scanner sc = **new** Scanner(System.***in***);  **while**(**true**)  {  System.***out***.println("Do u want to enter item details. Enter yes/no");  String v1 = sc.nextLine();  **if**(v1.equals("yes"))  {  System.***out***.println("Enter item id");  **int** id = Integer.*parseInt*(sc.nextLine());  System.***out***.println("Enter item name");  String item\_name = sc.nextLine();  System.***out***.println("Enter item cost");  **float** cost =Float.*parseFloat*(sc.nextLine());  ItemDetails it = **new** ItemDetails(id,item\_name,cost);  *ar\_items*.add(it);  }  **else**  {  **break**;  }  }  UserLogin ul = **new** UserLogin();      }  }  **public** **class** ItemDetails {  **int** item\_id;  String item\_name;  **float** cost;  ItemDetails(**int** it,String iname,**float** c)  {  item\_id =it;  item\_name = iname;  cost = c;  }  **public** String toString()  {  **return** item\_id+ " "+item\_name+ " "+cost+" ";  }  } | |
| 1. Write a program that uses Java Swing and JDBC to create a stand-alone application:   Create two tables namely, Representative (RepNo, RepName, State, Comission, Rate) and Customer (CustNo, CustName, State, Credit\_Limit, RepNo) in MySQL database.  Use appropriate Swing components to insert values in a form.  Use another form to display Representative’s information when 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();  }  } | |
| 1. 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> | |
|  | |
| 1. Write a JSP and Servlet Program to do the following to buy a T-Shirt online:    1. A set of checkboxes to select your T-Shirt accessories such as ‘belt’, ‘cap’, ‘hair-band’ etc.    2. A text area / text field to enter your T-Shirt tag-line    3. A Radio-button that allows the user to choose between T-Shirt with chest pocket and without.    4. A Combo Box to choose your T-Shirt color    5. Appropriate labels for these GUI Components    6. A Button called “Click Me” which when pressed will    7. Insert the details entered into a table called ‘TShirts’.    8. An OrderNo is generated by adding ‘1’ to the existing ‘OrderNo’    9. If ‘TShirts’ table is empty the initial value of ‘OrderNo’ is 100.    10. This ‘OrderNo’ is also inserted into the ‘TShirts’ table    11. 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> | |
| 1. 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.    1. Database Name: OnlineDirectory    2. Table Design:   Table Name: Telephone\_Directory  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> | |
| 1. 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.    1. *Login.html* will call *dataCapture.jsp* that should do the following:       1. Use Java Collections to make a list of valid users and facilitate user login functionality.       2. Give a personalized Welcome message and display today’s date.       3. Have a Text Entry with label ‘Name’ to enter the name of the user.       4. Give a List of Organizations to choose ‘Place of Work’       5. Provide a Male or Female option to choose the ‘Gender’       6. Have a Text Entry with label ‘Annual Income’       7. Give a Submit button reading ‘Calculate Tax’    2. *CalculateTax.jsp* must calculate the interest based on the following business rules:       1. Salary below 1,00,000 shall no have income-tax.       2. Calculate 15% of tax on 1,00,001 – 5,00,000.       3. Calculate 20% on 5,00,001 onwards.   PS: 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> | |
| 1. Create two tables Flight (Flight\_Number, Airline\_Name, Weekdays) and seatReservation(Flight\_Number, Date, Seat\_Number, Customer\_Name, Customer\_Phone) in MySQL database.   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.  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> | |
| 1. a. 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.   b.Write a program using java.net to transfer file from client socket to server socket.  **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();  }  }  b. //server program  **import** java.io.\*;  **import** java.net.\*;  **public** **class** MyServer {  **public** **static** **void** main(String[] args){  **try**{  ServerSocket ss=**new** ServerSocket(6666);  Socket s=ss.accept();//establishes connection  DataInputStream dis=**new** DataInputStream(s.getInputStream());  String  str=(String)dis.readUTF();  System.out.println("message= "+str);  ss.close();  }**catch**(Exception e){System.out.println(e);}  }  }  //client program  **import** java.io.\*;  **import** java.net.\*;  **public** **class** MyClient {  **public** **static** **void** main(String[] args) {  **try**{  Socket s=**new** Socket("localhost",6666);  DataOutputStream dout=**new** DataOutputStream(s.getOutputStream());  dout.writeUTF("Hello Server");  dout.flush();  dout.close();  s.close();  }**catch**(Exception e){System.out.println(e);}  }  } | |
| **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. | |
|  | |
| **Reference:**  1. Jim Keogh,The Complete Reference J2EE‟, TATA McGRAW-HILL Edition 2002.  2. B V Kumar, S Sangeetha, S V Subrahmanya, J2EE Architecture, TATA McGRAW-HILL Edition 2007. | |
|  | |
| **Course Outcomes:**  At the end of the course, students will be able to | |
| **1.** | Develop programs using collections such as lists, hashmaps and sets. (PO-1(2), 2(2), 3(3),5(3)) (PSO-1(3),2(3)) |
| **2.** | Develop front ends using swings and manipulate stored data. (PO-1(2), 2(2), 3(3),5(3)) (PSO-1(3),2(3)) |
| **3.** | Develop web applications using servlets, jsps and manipulate the data. (PO- 1(2), 2(2), 3(3),5(3)) (PSO-1(3),2(3)) |
| **4.** | Perform concurrent execution of programs to improve CPU utilization. (PO-1(2), 2(2), 3(3),5(3)) (PSO-1(3),2(3)) |
| **5.** | Develop reports and communicate effectively on the concepts related to Java and J2EE. (PO-1(1),2(2), 10(3)) (PSO-2(2)) |