**Implementation:**

**File: Main.java**

**class Employee {**

**private String firstName;**

**private String lastName;**

**private String ID;**

**private double basicSalary, incRate;**

**protected Employee(String fname, String lname, String id, double bs, double inc) {**

**firstName = fname;**

**lastName = lname;**

**ID = id;**

**basicSalary = bs;**

**incRate = inc;**

**}**

**String getFirstName() {**

**return firstName;**

**}**

**String getLastName() {**

**return lastName;**

**}**

**String getID() {**

**return ID;**

**}**

**double getBasicSalary() {**

**return basicSalary;**

**}**

**double getIncRate() {**

**return incRate;**

**}**

**}**

**class HouseRent extends Employee {**

**private double houseRentRate;**

**protected HouseRent(String fname, String lname, String id,**

**double bs, double inc, double hRentRate) {**

**super(fname, lname, id, bs, inc);**

**houseRentRate = hRentRate;**

**}**

**double getHouseRentRate() {**

**return houseRentRate;**

**}**

**}**

**class ProvidendFund extends HouseRent {**

**double grossSalary;**

**double providendFundRate;**

**protected ProvidendFund(String fname, String lname, String id,**

**double bs, double inc, double hRentRate,**

**double pFundRate) {**

**super(fname, lname, id, bs, inc, hRentRate);**

**providendFundRate = pFundRate;**

**}**

**double calculateGrossSalary() {**

**double bSalary = getBasicSalary();**

**grossSalary = bSalary + bSalary \* getIncRate();**

**grossSalary += (bSalary \* getHouseRentRate() / 100);**

**grossSalary \*= 12;**

**return grossSalary;**

**}**

**double calculateNetSalary() {**

**calculateGrossSalary();**

**double netSalary = grossSalary;**

**netSalary -= 12 \* (getBasicSalary() \* providendFundRate / 100);**

**return netSalary;**

**}**

**void printGrossAndNetSalary() {**

**System.out.println("Employee Yearly Salary Info: ");**

**System.out.println("Name: " + getFirstName() + " " + getLastName());**

**System.out.println("Employee ID: " + getID());**

**System.out.println("Basic Salary(per month): " + getBasicSalary());**

**System.out.println("Increment Rate(per year): " + getIncRate() + "%");**

**System.out.println("House Rent(per month): " +**

**(getBasicSalary() \* getHouseRentRate() / 100));**

**System.out.println("Providend Fund(per month): " +**

**(getBasicSalary() \* providendFundRate / 100));**

**System.out.println("Gross Salary(in a year): " + calculateGrossSalary());**

**System.out.println("Net Salary(in a year): " + calculateNetSalary());**

**}**

**}**

**class Main {**

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

**ProvidendFund p = new ProvidendFund("sharif", "omar", "1969dhaka",**

**50000, 4.0, 10.0, 12.0);**

**p.printGrossAndNetSalary();**

**}**

**}**

**Experiment: 9**

**Implementation:**

**File:Main.java**

**abstract class Dog {**

**void walk() {**

**System.out.println("dog walking...");**

**}**

**void sleep() {**

**System.out.println("dog sleeping...");**

**}**

**void bark() {**

**System.out.println("dog barking...");**

**}**

**abstract void eat();**

**}**

**class AmericanDog extends Dog {**

**void eat() {**

**System.out.println("American dog eating...");**

**}**

**}**

**class ChineseDog extends Dog {**

**void eat() {**

**System.out.println("Chinese dog eating...");**

**}**

**}**

**class AustralianDog extends Dog {**

**void eat() {**

**System.out.println("Australian dog eating...");**

**}**

**}**

**class CanadianDog extends Dog {**

**void eat() {**

**System.out.println("Canadian dog eating...");**

**}**

**}**

**class IndianDog extends Dog {**

**void eat() {**

**System.out.println("Indian dog eating...");**

**}**

**}**

**class HungarianDog extends Dog {**

**void eat() {**

**System.out.println("Hungarian dog eating...");**

**}**

**}**

**class Main {**

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

**Dog d = new AmericanDog();**

**d.eat();**

**d = new ChineseDog();**

**d.eat();**

**d = new AustralianDog();**

**d.eat();**

**d = new CanadianDog();**

**d.eat();**

**d = new IndianDog();**

**d.eat();**

**d = new HungarianDog();**

**d.eat();**

**d.walk();**

**d.bark();**

**d.sleep();**

**}**

**}**

**Experiment: 10**

**Implementation:**

**File:Main.java**

**import java.util.Scanner;**

**interface Vehicle {**

**public void changeGear();**

**public void speedUp();**

**public void speedDown();**

**public void pressBreak();**

**}**

**class Bus implements Vehicle {**

**public void changeGear() {**

**System.out.println("Bus");**

**System.out.println("Changing gear...");**

**}**

**public void speedUp() {**

**System.out.println("Bus");**

**System.out.println("Speeding up...");**

**}**

**public void speedDown() {**

**System.out.println("Bus");**

**System.out.println("Speeding down...");**

**}**

**public void pressBreak() {**

**System.out.println("Bus");**

**System.out.println("Pressed Break...");**

**}**

**}**

**class Truck implements Vehicle {**

**public void changeGear() {**

**System.out.println("Truck");**

**System.out.println("Changing gear...");**

**}**

**public void speedUp() {**

**System.out.println("Truck");**

**System.out.println("Speeding up...");**

**}**

**public void speedDown() {**

**System.out.println("Truck");**

**System.out.println("Speeding down...");**

**}**

**public void pressBreak() {**

**System.out.println("Truck");**

**System.out.println("Pressed Break...");**

**}**

**}**

**class Motorcycle implements Vehicle {**

**public void changeGear() {**

**System.out.println("Motorcycle");**

**System.out.println("Changing gear...");**

**}**

**public void speedUp() {**

**System.out.println("Motorcycle");**

**System.out.println("Speeding up...");**

**}**

**public void speedDown() {**

**System.out.println("Motorcycle");**

**System.out.println("Speeding down...");**

**}**

**public void pressBreak() {**

**System.out.println("Motorcycle");**

**System.out.println("Pressed Break...");**

**}**

**}**

**class Microbus implements Vehicle {**

**public void changeGear() {**

**System.out.println("Microbus");**

**System.out.println("Changing gear...");**

**}**

**public void speedUp() {**

**System.out.println("Microbus");**

**System.out.println("Speeding up...");**

**}**

**public void speedDown() {**

**System.out.println("Microbus");**

**System.out.println("Speeding down...");**

**}**

**public void pressBreak() {**

**System.out.println("Microbus");**

**System.out.println("Pressed Break...");**

**}**

**}**

**class CNG implements Vehicle {**

**public void changeGear() {**

**System.out.println("CNG");**

**System.out.println("Changing gear...");**

**}**

**public void speedUp() {**

**System.out.println("CNG");**

**System.out.println("Speeding up...");**

**}**

**public void speedDown() {**

**System.out.println("CNG");**

**System.out.println("Speeding down...");**

**}**

**public void pressBreak() {**

**System.out.println("CNG");**

**System.out.println("Pressed Break...");**

**}**

**}**

**class Main {**

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

**System.out.println("Choose an option:\n1. Bus\n2. Truck\n3. Motorcycle\n4. Microbus\n5. CNG");**

**Scanner sc = new Scanner(System.in);**

**int option = sc.nextInt();**

**Vehicle vehicle = new Bus();**

**switch (option) {**

**case 1:**

**vehicle = new Bus();**

**break;**

**case 2:**

**vehicle = new Truck();**

**break;**

**case 3:**

**vehicle = new Motorcycle();**

**break;**

**case 4:**

**vehicle = new Microbus();**

**break;**

**case 5:**

**vehicle = new CNG();**

**break;**

**default:**

**System.out.println("Wrong Option!\n");**

**System.exit(-1);**

**}**

**vehicle.changeGear();**

**vehicle.speedUp();**

**vehicle.speedDown();**

**vehicle.pressBreak();**

**sc.close();**

**}**

**}**

**Experiment: 11**

**Implementation:**

**File: Main.java**

**import examcontroller.ExamController;**

**import registrar.RegistrarOffice;**

**public class Main {**

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

**RegistrarOffice rof = new RegistrarOffice();**

**rof.recruitTeacher();**

**rof.recruitOfficer();**

**rof.recruitStaff();**

**rof.enrollStudents();**

**ExamController ecnt = new ExamController();**

**ecnt.publishAdmissionTestResult();**

**ecnt.processResults();**

**}**

**}**

**File: RegisterOffice.java**

**package registrar;**

**interface Recruitment {**

**public void recruit();**

**}**

**class Teacher implements Recruitment {**

**public void recruit() {**

**viva();**

**}**

**private void viva() {**

**System.out.println("Taking viva for teacher recruitment...");**

**System.out.println("Viva Completed...");**

**}**

**}**

**class Officer implements Recruitment {**

**public void recruit() {**

**writtenExam();**

**viva();**

**}**

**private void writtenExam() {**

**System.out.println("Taking written exam for officer recruitment...");**

**System.out.println("Written Exam completed...");**

**}**

**private void viva() {**

**System.out.println("Taking viva for officer recruitment...");**

**System.out.println("Viva Completed...");**

**}**

**}**

**class Staff implements Recruitment {**

**public void recruit() {**

**writtenExam();**

**practicalExam();**

**viva();**

**}**

**private void writtenExam() {**

**System.out.println("Taking written exam for staff recruitment...");**

**System.out.println("Written Exam completed...");**

**}**

**private void viva() {**

**System.out.println("Taking viva for staff recruitment...");**

**System.out.println("Viva Completed...");**

**}**

**private void practicalExam() {**

**System.out.println("Taking pratical exam for Staff recruitment...");**

**System.out.println("Practical Exam Completed...");**

**}**

**}**

**class Students {**

**public void takeAdmissionTest() {**

**System.out.println("Taking admission test for student enrollment...");**

**System.out.println("Admission Test Completed...");**

**}**

**}**

**public class RegistrarOffice {**

**Recruitment r;**

**public void recruitTeacher() {**

**r = new Teacher();**

**r.recruit();**

**}**

**public void recruitOfficer() {**

**r = new Officer();**

**r.recruit();**

**}**

**public void recruitStaff() {**

**r = new Staff();**

**r.recruit();**

**}**

**public void enrollStudents() {**

**Students s = new Students();**

**s.takeAdmissionTest();**

**}**

**}**

**File: ExamController.java**

**package examcontroller;**

**import accounts.Accounts;**

**interface Result {**

**public void publishResult();**

**}**

**class Science implements Result {**

**public void publishResult() {**

**System.out.println("Publishing science faculty result...");**

**}**

**}**

**class Engineering implements Result {**

**public void publishResult() {**

**System.out.println("Publishing Engineering faculty result...");**

**}**

**}**

**class LifeScience implements Result {**

**public void publishResult() {**

**System.out.println("Publishing Life Science faculty result...");**

**}**

**}**

**class SocialScience implements Result {**

**public void publishResult() {**

**System.out.println("Publishing Social Science faculty result...");**

**}**

**}**

**class Arts implements Result {**

**public void publishResult() {**

**System.out.println("Publishing Arts faculty result...");**

**}**

**}**

**class BusinessAdministration implements Result {**

**public void publishResult() {**

**System.out.println("Publishing Business Administration faculty result...");**

**}**

**}**

**class Law implements Result {**

**public void publishResult() {**

**System.out.println("Publishing Law faculty result...");**

**}**

**}**

**public class ExamController {**

**private Accounts ac = new Accounts();**

**public void processResults() {**

**System.out.println("Result processing going on...");**

**ac.haveStudentsPaidExamFees();**

**System.out.println("Processing results of 7 faculties...");**

**Result r = new Science();**

**r.publishResult();**

**r = new Engineering();**

**r.publishResult();**

**r = new LifeScience();**

**r.publishResult();**

**r = new SocialScience();**

**r.publishResult();**

**r = new Arts();**

**r.publishResult();**

**r = new BusinessAdministration();**

**r.publishResult();**

**r = new Law();**

**r.publishResult();**

**}**

**public void publishAdmissionTestResult() {**

**System.out.println("Publishing admission test result...");**

**}**

**}**

**File: Accounts.java**

**package accounts;**

**public class Accounts {**

**private void areStudentsValid() {**

**System.out.println("#A Students are valid and #B students are invalid.");**

**}**

**public void haveStudentsPaidExamFees() {**

**areStudentsValid();**

**System.out.println("Checking for exam payments...");**

**System.out.println("#A Students have paid exam fees and #B students have not paid exam fees.");**

**}**

**}**

**Experiment: 12**

**Implementation:**

**package main;**

**import java.util.Scanner;**

**import student.Student;**

**public class Main {**

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

**System.out.println("What's your ID? ");**

**Scanner sc = new Scanner(System.in);**

**String ID = sc.next();**

**Student s = new Student(ID);**

**s.collectForm();**

**s.submitForm();**

**sc.close();**

**}**

**}**

**File: Form.java**

**package form;**

**public class Form {**

**private String ID;**

**private int fees;**

**public void setID(String id) {**

**ID = id;**

**}**

**public void setFees(int f) { fees = f; }**

**public int getFees() { return fees; }**

**public String getID() { return ID; }**

**}**

**File: Student.java**

**package student;**

**import cseoffice.CseOffice;**

**import form.Form;**

**public class Student {**

**private String ID;**

**Form f;**

**CseOffice cseOffice = new CseOffice();**

**public Student(String id) {**

**ID = id;**

**}**

**public void collectForm() {**

**f = cseOffice.getForm();**

**f.setID(ID);**

**}**

**public void submitForm() {**

**try {**

**cseOffice.submitForm(f);**

**} catch (Exception e) {**

**System.out.println(e);**

**}**

**}**

**}**

**File: CseOffice.java**

**package cseoffice;**

**import form.Form;**

**import registraroffice.RegistrarOffice;**

**import java.lang.Math;**

**import exceptions.HasFailRecord;**

**import exceptions.HasFeesDue;**

**import exceptions.LowAttendance;**

**import exceptions.PaidLess;**

**public class CseOffice {**

**private RegistrarOffice regOffice = new RegistrarOffice();**

**public Form getForm() {**

**return new Form();**

**}**

**public void submitForm(Form f) throws LowAttendance, HasFailRecord, HasFeesDue, PaidLess {**

**// Using ID's hash code to generate random attendance**

**int ID = Math.abs(f.getID().hashCode());**

**int attendance = getAttendance(ID);**

**if (attendance < 70) {**

**throw new LowAttendance();**

**}**

**regOffice.submitForm(f);**

**System.out.println("Admit card for ID: " + f.getID() + " has been collected.");**

**}**

**public int getAttendance(int ID) {**

**// generating a random attendance between 60 to 100**

**return (int)(Math.random() \* ID) % 100 + 60;**

**}**

**}**

**File: RegisterOffice.java**

**package registraroffice;**

**import java.lang.Math;**

**import accounts.AccountsOffice;**

**import exceptions.HasFailRecord;**

**import exceptions.HasFeesDue;**

**import exceptions.PaidLess;**

**import form.Form;**

**public class RegistrarOffice {**

**AccountsOffice acOffice = new AccountsOffice();**

**public void submitForm(Form f) throws HasFailRecord, HasFeesDue, PaidLess {**

**int ID = Math.abs(f.getID().hashCode());**

**if (checkIfFailed(ID)) {**

**throw new HasFailRecord();**

**}**

**acOffice.calcuateFees(f);**

**}**

**private boolean checkIfFailed(int ID) {**

**// Generating random number between 0 and 1**

**int r = ((int)(Math.random() \* (double)ID) + 5) % 2;**

**return (r == 1 ? false : true);**

**}**

**}**

**File: AccountsOffice.java**

**package accounts;**

**import bank.Bank;**

**import exceptions.HasFeesDue;**

**import exceptions.PaidLess;**

**import form.Form;**

**public class AccountsOffice {**

**Bank bank = new Bank();**

**public void calcuateFees(Form f) throws HasFeesDue, PaidLess {**

**if (hasDue(Math.abs(f.hashCode()))) {**

**throw new HasFeesDue();**

**}**

**f.setFees(fees());**

**bank.takeFees(f);**

**}**

**private int fees() {**

**return 1560;**

**}**

**private boolean hasDue(int ID) {**

**int r = ((int)(Math.random() \* (double)ID)+5) % 2;**

**return (r == 1 ? false : true);**

**}**

**}**

**File: Bank.java**

**package bank;**

**import form.Form;**

**import java.util.Scanner;**

**import examcontroller.ExamController;**

**import exceptions.PaidLess;**

**public class Bank {**

**ExamController eController = new ExamController();**

**public void takeFees(Form f) throws PaidLess {**

**int fees = f.getFees();**

**System.out.println("Pay " + fees + "tk: ");**

**Scanner sc = new Scanner(System.in);**

**int paidFees = sc.nextInt();**

**if (paidFees < fees) {**

**throw new PaidLess();**

**}**

**eController.generateAdmitCard(f);**

**}**

**}**

**File: ExamController.java**

**package examcontroller;**

**import form.Form;**

**public class ExamController {**

**public void generateAdmitCard(Form f) {**

**System.out.println("Admit card has been generated for ID: " + f.getID());**

**System.out.println("Admit card has been sent to CSE office.");**

**}**

**}**

**File: LowAttendence.java**

**package exceptions;**

**import java.lang.Exception;**

**public class LowAttendance extends Exception {**

**public String toString() {**

**return "attendance less than 70%";**

**}**

**}**

**File; HasFailRecord.java**

**package exceptions;**

**import java.lang.Exception;**

**public class HasFailRecord extends Exception {**

**public String toString() {**

**return "the student has previously failed in this exam";**

**}**

**}**

**File: HasFeesDue.java**

**package exceptions;**

**import java.lang.Exception;**

**public class HasFeesDue extends Exception {**

**public String toString() {**

**return "student has due fees";**

**}**

**}**

**File: PaidLess.java**

**package exceptions;**

**import java.lang.Exception;**

**public class PaidLess extends Exception {**

**public String toString() {**

**return "student paid less than what was asked";**

**}**

**}**

**Experiment: 13**

**Implementation:**

**File: Main.java**

**package exp13;**

**public class Main {**

**public static void main(String[] args) throws InterruptedException {**

**VcOffice vcOffice = new VcOffice();**

**vcOffice.askForNewDeptNames();**

**}**

**}**

**File: VcOffice.java**

**package exp13;**

**abstract class Faculty implements Runnable {**

**abstract public void performMeeting();**

**public void run() {**

**performMeeting();**

**}**

**}**

**class Enggineering extends Faculty {**

**public void performMeeting() {**

**int n = (int)(Math.random() \* 100000000) % 100000;**

**String s = new String();**

**while (n > 0) {**

**s += (char)(n%10 + 'a');**

**n /= 10;**

**}**

**System.out.println("Engineering faculty has proposed '" + s + "' as new departname.");**

**}**

**}**

**class Science extends Faculty {**

**public void performMeeting() {**

**int n = (int)(Math.random() \* 100000000) % 100000;**

**String s = new String();**

**while (n > 0) {**

**s += (char)(n%10 + 'a');**

**n /= 10;**

**}**

**System.out.println("Science faculty has proposed '" + s + "' as new departname.");**

**}**

**}**

**class Arts extends Faculty {**

**public void performMeeting() {**

**int n = (int)(Math.random() \* 100000000) % 100000;**

**String s = new String();**

**while (n > 0) {**

**s += (char)(n%10 + 'a');**

**n /= 10;**

**}**

**System.out.println("Arts faculty has proposed '" + s + "' as new departname.");**

**}**

**}**

**class Law extends Faculty {**

**public void performMeeting() {**

**int n = (int)(Math.random() \* 100000000) % 100000;**

**String s = new String();**

**while (n > 0) {**

**s += (char)(n%10 + 'a');**

**n /= 10;**

**}**

**System.out.println("Law faculty has proposed '" + s + "' as new departname.");**

**}**

**}**

**class SocialScience extends Faculty {**

**public void performMeeting() {**

**int n = (int)(Math.random() \* 100000000) % 100000;**

**String s = new String();**

**while (n > 0) {**

**s += (char)(n%10 + 'a');**

**n /= 10;**

**}**

**System.out.println("Social Science faculty has proposed '" + s + "' as new departname.");**

**}**

**}**

**class BusinessAdministration extends Faculty {**

**public void performMeeting() {**

**int n = (int)(Math.random() \* 100000000) % 100000;**

**String s = new String();**

**while (n > 0) {**

**s += (char)(n%10 + 'a');**

**n /= 10;**

**}**

**System.out.println("Business Administration faculty has proposed '" + s + "' as new departname.");**

**}**

**}**

**class LifeScience extends Faculty {**

**public void performMeeting() {**

**int n = (int)(Math.random() \* 100000000) % 100000;**

**String s = new String();**

**while (n > 0) {**

**s += (char)(n%10 + 'a');**

**n /= 10;**

**}**

**System.out.println("Life Sciencde faculty has proposed '" + s + "' as new departname.");**

**}**

**}**

**public class VcOffice {**

**Faculty faculties[] = new Faculty[] {**

**new Enggineering(),**

**new Science(),**

**new Arts(),**

**new Law(),**

**new SocialScience(),**

**new BusinessAdministration(),**

**new LifeScience()**

**};**

**public void askForNewDeptNames() throws InterruptedException {**

**Thread t;**

**for (int i = 0; i < 7; ++i) {**

**t = new Thread(faculties[i]);**

**t.start();**

**t.join();**

**}**

**finalizeAllNewDepts();**

**}**

**public void finalizeAllNewDepts() {**

**System.out.println("All new department proposals have been finalized.");**

**}**

**}**

**Experiment: 14**

**Implementation:**

**File: Main.java**

**package exp14;**

**public class Main {**

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

**new Thread(new WordCounter("Thread1", 33)).start();**

**new Thread(new WordCounter("Thread2", 33)).start();**

**new Thread(new WordCounter("Thread3", 34)).start();**

**}**

**}**

**File: WordCounter.java**

**package exp14;**

**import java.io.IOException;**

**public class WordCounter implements Runnable {**

**private static CountWords countWords;**

**private int paragraphToBeRead;**

**private String tName;**

**public WordCounter(String name, int n) {**

**paragraphToBeRead = n;**

**tName = name;**

**}**

**static {**

**try {**

**countWords = new CountWords("C:\\Users\\Dell\\eclipse-workspace\\Exp14\\src\\paragraph.txt");**

**} catch (IOException e) {**

**e.printStackTrace();**

**}**

**}**

**public void run() {**

**int n = ((countWords.getParagraphSize() \* paragraphToBeRead) / 100);**

**int wCount = 0;**

**for (int i = 0; i < n; ++i) {**

**try {**

**String s = countWords.readNextWord();**

**if (s.isEmpty())**

**break;**

**i += s.length();**

**++wCount;**

**} catch (IOException e) {**

**e.printStackTrace();**

**}**

**}**

**System.out.println("'" + tName + "' thread counted: " + wCount + " words.");**

**}**

**}**

**File: CountWords.java**

**package exp14;**

**import java.io.FileInputStream;**

**import java.io.IOException;**

**public class CountWords {**

**private String text;**

**int textIndex = 0;**

**public CountWords(String fname) throws IOException {**

**FileInputStream is = new FileInputStream(fname);**

**byte[] b = is.readAllBytes();**

**text = new String(b);**

**is.close();**

**}**

**public int getParagraphSize() {**

**return text.length();**

**}**

**public synchronized String readNextWord() throws IOException {**

**if (textIndex >= text.length())**

**return new String("");**

**while (textIndex < text.length() &&**

**Character.isWhitespace(text.charAt(textIndex))) {**

**++textIndex;**

**}**

**if (textIndex >= text.length())**

**return new String("");**

**int startIndex = textIndex;**

**while (textIndex < text.length() &&**

**!Character.isWhitespace(text.charAt(textIndex)))**

**++textIndex;**

**return text.substring(startIndex, textIndex);**

**}**

**}**

**Experiment: 15**

**Implementation:**

**File: Main.java**

**import java.awt.Color;**

**import java.awt.EventQueue;**

**import javax.swing.JFrame;**

**import javax.swing.JPanel;**

**import javax.swing.border.EmptyBorder;**

**import exp15.SaveInfo;**

**import javax.swing.JLabel;**

**import javax.swing.JOptionPane;**

**import java.awt.Font;**

**import javax.swing.JTextField;**

**import javax.swing.JButton;**

**import java.awt.event.ActionListener;**

**import java.io.IOException;**

**import java.awt.event.ActionEvent;**

**public class MainWindow extends JFrame {**

**private JPanel contentPane;**

**private JTextField textField;**

**private JTextField textField\_1;**

**private JTextField textField\_2;**

**private SaveInfo sInfo;**

**/\*\***

**\* Launch the application.**

**\*/**

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

**EventQueue.invokeLater(new Runnable() {**

**public void run() {**

**try {**

**MainWindow frame = new MainWindow();**

**frame.setVisible(true);**

**} catch (Exception e) {**

**e.printStackTrace();**

**}**

**}**

**});**

**}**

**public MainWindow() throws IOException {**

**sInfo = new SaveInfo("C:\\Users\\Dell\\eclipse-workspace\\Exp15\\src\\employee.dat", true);**

**setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);**

**setBounds(100, 100, 450, 300);**

**contentPane = new JPanel();**

**contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));**

**setContentPane(contentPane);**

**contentPane.setLayout(null);**

**JLabel lblNewLabel = new JLabel("Name");**

**lblNewLabel.setFont(new Font("Segoe UI", Font.PLAIN, 13));**

**lblNewLabel.setBounds(26, 45, 60, 25);**

**contentPane.add(lblNewLabel);**

**textField = new JTextField();**

**textField.setFont(new Font("Segoe UI", Font.PLAIN, 13));**

**textField.setBounds(112, 44, 314, 26);**

**contentPane.add(textField);**

**textField.setColumns(10);**

**JLabel lblDesignation = new JLabel("Designation");**

**lblDesignation.setFont(new Font("Segoe UI", Font.PLAIN, 13));**

**lblDesignation.setBounds(26, 81, 76, 25);**

**contentPane.add(lblDesignation);**

**textField\_1 = new JTextField();**

**textField\_1.setFont(new Font("Segoe UI", Font.PLAIN, 13));**

**textField\_1.setColumns(10);**

**textField\_1.setBounds(112, 80, 314, 26);**

**contentPane.add(textField\_1);**

**JLabel lblBasicSalary = new JLabel("Basic Salary");**

**lblBasicSalary.setFont(new Font("Segoe UI", Font.PLAIN, 13));**

**lblBasicSalary.setBounds(26, 117, 76, 25);**

**contentPane.add(lblBasicSalary);**

**textField\_2 = new JTextField();**

**textField\_2.setFont(new Font("Segoe UI", Font.PLAIN, 13));**

**textField\_2.setColumns(10);**

**textField\_2.setBounds(112, 116, 314, 26);**

**contentPane.add(textField\_2);**

**JButton btnNewButton = new JButton("OK");**

**btnNewButton.addActionListener(new ActionListener() {**

**public void actionPerformed(ActionEvent e) {**

**String n = textField.getText();**

**String d = textField\_1.getText();**

**String s = textField\_2.getText();**

**if (n.isEmpty() || d.isEmpty() || s.isEmpty()) {**

**JOptionPane.showMessageDialog(btnNewButton, "Please make sure all the fields are filled with information!");**

**return;**

**}**

**textField.setText("");**

**textField\_1.setText("");**

**textField\_2.setText("");**

**try {**

**sInfo.write("Name: " + n + "\n") ;**

**sInfo.write("Designation: " + d + "\n");**

**sInfo.write("Basic Salary: " + s + "\n");**

**} catch (IOException e1) {**

**e1.printStackTrace();**

**}**

**}**

**});**

**btnNewButton.setFont(new Font("Segoe UI", Font.PLAIN, 13));**

**btnNewButton.setBounds(230, 220, 85, 21);**

**contentPane.add(btnNewButton);**

**btnNewButton.setBackground(new Color(0xff, 0xff, 0xff));**

**JButton btnCancel = new JButton("Cancel");**

**btnCancel.addActionListener(new ActionListener() {**

**public void actionPerformed(ActionEvent e) {**

**try {**

**sInfo.close();**

**} catch (Exception excep) {**

**excep.printStackTrace();**

**}**

**System.exit(0);**

**}**

**});**

**btnCancel.setFont(new Font("Segoe UI", Font.PLAIN, 13));**

**btnCancel.setBounds(325, 220, 85, 21);**

**btnCancel.setBackground(new Color(0xff, 0xff, 0xff));**

**contentPane.add(btnCancel);**

**textField.addActionListener(new ActionListener() {**

**public void actionPerformed(ActionEvent e) {**

**String n = textField.getText();**

**String d = textField\_1.getText();**

**String s = textField\_2.getText();**

**if (n.isEmpty() || d.isEmpty() || s.isEmpty()) {**

**JOptionPane.showMessageDialog(textField, "Please make sure all the fields are filled with information!");**

**return;**

**}**

**btnNewButton.doClick();**

**}**

**});**

**textField\_1.addActionListener(new ActionListener() {**

**public void actionPerformed(ActionEvent e) {**

**String n = textField.getText();**

**String d = textField\_1.getText();**

**String s = textField\_2.getText();**

**if (n.isEmpty() || d.isEmpty() || s.isEmpty()) {**

**JOptionPane.showMessageDialog(textField\_1, "Please make sure all the fields are filled with information!");**

**return;**

**}**

**btnNewButton.doClick();**

**}**

**});**

**textField\_2.addActionListener(new ActionListener() {**

**public void actionPerformed(ActionEvent e) {**

**String n = textField.getText();**

**String d = textField\_1.getText();**

**String s = textField\_2.getText();**

**if (n.isEmpty() || d.isEmpty() || s.isEmpty()) {**

**JOptionPane.showMessageDialog(textField\_2, "Please make sure all the fields are filled with information!");**

**return;**

**}**

**btnNewButton.doClick();**

**}**

**});**

**}**

**}**

**File: SaveInfo.java**

**package exp15;**

**import java.io.FileWriter;**

**import java.io.IOException;**

**public class SaveInfo {**

**FileWriter fw;**

**public SaveInfo(String fName, boolean append) throws IOException {**

**fw = new FileWriter(fName, append);**

**}**

**public void write(String s) throws IOException {**

**fw.write(s);**

**}**

**public void close() throws IOException {**

**fw.close();**

**}**

**}**