**Main Method()**

*public* *class* GUI{

*protected* *static* *GUI2* f2 = new GUI2(1280, 720);

*protected* *static* *GUI3* f3 = new GUI3(800, 800);

*protected* *static* *GUISearch* f4 = new GUISearch(800, 500);

*protected* *static* update f5 = new update(1000, 400);

*protected* *static* delete f6 = new delete(800, 500);

*protected* *static* *FileOperations* f1 = new FileOperations();

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

        department d1 = new department("CS", "ISB");

        student s1 = new student("hafif", "123456789101", "Male", "BCS 3A", 4, 3, d1);

        student s2 = new student("Sam", "123456789101", "Male", "BCS 3A", 4, 3, d1);

        student s3 = new student("Mansoor", "123456789101", "Male", "BCS 3A", 4, 3, d1);

*/\* f1.writeToFile(d1, "department");*

*f1.writeToFile(s1, "student");*

*f1.writeToFile(s2, "student");*

*f1.writeToFile(s3, "student"); \*/*

        f1.readFromFile("student");

        f1.readFromFile("department");

        f2.addButtons();

        f2.setVisible(true);

    }

}

**Question 1 (Home Screen):**

*public* *class* GUI2{

*private* *JFrame* frame;

*// Argumented constructor*

*public* GUI2(*int* *width*, *int* *height*){

*this*.frame = new JFrame();

*// Essentials for menu display*

*this*.frame.setSize(width, height);

*this*.frame.setLayout(null);

*this*.frame.setResizable(false);

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

*this*.frame.getContentPane().setBackground(Color.GRAY);

    }

*public* *JFrame* getFrame() {

        return frame;

    }

*// setter*

*public* *void* setVisible(*boolean* *check*){

*this*.frame.setVisible(check);

    }

*// Buttons*

*public* *void* addButtons(){

*// Buttons*

        buttons b3 = new buttons();

*this*.frame.add(b3.newButton(frame, 300, 100, 200, 100, "Add"));

        buttons b4 = new buttons();

*this*.frame.add(b4.newButton(frame, 300, 300, 200, 100, "Search"));

        buttons b5 = new buttons();

*this*.frame.add(b5.newButton(frame, 300, 500, 200, 100, "Update"));

        buttons b6 = new buttons();

*this*.frame.add(b6.newButton(frame, 550, 100, 200, 100, "Delete"));

        buttons b7 = new buttons();

*this*.frame.add(b7.newButton(frame, 550, 300, 200, 100, "Display"));

    }

}

*class* textFields{

*JTextField* field;

*JLabel* label;

*public* textFields(){

    }

*public* *JTextField* createFields(*JFrame* *frame*, *int* *x*, *int* *y*, *int* *width*, *int* *height*, *String* *text*){

*this*.field = new JTextField(20);

*this*.label = new JLabel();

*this*.field.setBounds(x, y, width, height);;

*this*.label.setText(text);

*this*.label.setForeground(Color.WHITE);

*this*.label.setBounds(x - 150, y, width, height);

        frame.add(label);

        frame.add(field);

        return field;

    }

*public* *String* getFieldText(){

        return field.getText();

    }

}

**Question 2 (Add):**

*public* *class* GUI3{

*private* *JFrame* frame;

*protected* *static* textFields t1, t2, t3, t4, t5, t6, t7, t8;

*// Argumented constructor*

*public* GUI3(*int* *width*, *int* *height*){

*this*.frame = new JFrame();

*// Essentials for menu display*

*this*.frame.setSize(width, height);

*this*.frame.setLayout(null);

*this*.frame.setResizable(false);

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

*this*.frame.getContentPane().setBackground(Color.GRAY);

*// TextFields*

        t1 = new textFields();

*this*.frame.add(t1.createFields(frame, 300, 50, 200, 50, "NAME"));

        t2 = new textFields();

*this*.frame.add(t2.createFields(frame, 300, 120, 200, 50, "PHONE"));

        t3 = new textFields();

*this*.frame.add(t3.createFields(frame, 300, 190, 200, 50, "GENDER"));

        t4 = new textFields();

*this*.frame.add(t4.createFields(frame, 300, 260, 200, 50, "DEPT NAME"));

        t5 = new textFields();

*this*.frame.add(t5.createFields(frame, 300, 330, 200, 50, "DEPT LOCATION"));

        t6 = new textFields();

*this*.frame.add(t6.createFields(frame, 300, 400, 200, 50, "GPA"));

        t7 = new textFields();

*this*.frame.add(t7.createFields(frame, 300, 470, 200, 50, "SEMESTER"));

        t8 = new textFields();

*this*.frame.add(t8.createFields(frame, 300, 540, 200, 50, "SECTION"));

    }

*// setter*

*public* *void* setVisible(*boolean* *check*){

*this*.frame.setVisible(check);

    }

*public* *JFrame* getFrame() {

        return frame;

    }

*// Buttons*

*public* *void* addButtons(){

*// Buttons*

        buttons b6 = new buttons();

*this*.frame.add(b6.newButton(frame, 150, 640, 200, 50, "SUBMIT"));

        buttons b7 = new buttons();

*this*.frame.add(b7.newButton(frame, 380, 640, 200, 50, "HOME"));

    }

}

**Question 3 (Search):**

*public* *class* GUISearch{

*private* *JFrame* frame;

*protected* textFields t9;

*// Argumented constructor*

*public* GUISearch(*int* *width*, *int* *height*){

*this*.frame = new JFrame();

*// Essentials for menu display*

*this*.frame.setSize(width, height);

*this*.frame.setLayout(null);

*this*.frame.setResizable(false);

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

*this*.frame.getContentPane().setBackground(Color.GRAY);

        t9 = new textFields();

*this*.frame.add(t9.createFields(frame, 300, 50, 200, 50, "NAME"));

    }

*// setter*

*public* *void* setVisible(*boolean* *check*){

*this*.frame.setVisible(check);

    }

*public* *JFrame* getFrame() {

        return frame;

    }

*// Buttons*

*public* *void* addButtons(){

*// Buttons*

        buttons b8 = new buttons();

*this*.frame.add(b8.newButton(frame, 150, 150, 200, 50, "SEARCH"));

        buttons b9 = new buttons();

*this*.frame.add(b9.newButton(frame, 380, 150, 200, 50, "HOME"));

    }

}

**Question 4 (Update):**

*public* *class* update {

*private* *JFrame* frame;

*protected* textFields t10, t11;

*// Argumented constructor*

*public* update(*int* *width*, *int* *height*){

*this*.frame = new JFrame();

*// Essentials for menu display*

*this*.frame.setSize(width, height);

*this*.frame.setLayout(null);

*this*.frame.setResizable(false);

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

*this*.frame.getContentPane().setBackground(Color.gray);

        t10 = new textFields();

*this*.frame.add(t10.createFields(frame, 300, 50, 200, 50, "OLD NAME"));

        t11 = new textFields();

*this*.frame.add(t11.createFields(frame, 700, 50, 200, 50, "NEW NAME"));

    }

*// setter*

*public* *void* setVisible(*boolean* *check*){

*this*.frame.setVisible(check);

    }

*public* *JFrame* getFrame() {

        return frame;

    }

*// Buttons*

*public* *void* addButtons(){

*// Buttons*

        buttons b10 = new buttons();

*this*.frame.add(b10.newButton(frame, 151, 150, 200, 50, "UPDATE"));

        buttons b11 = new buttons();

*this*.frame.add(b11.newButton(frame, 380, 150, 200, 50, "HOME"));

    }

}

**Class Writing:**

*class* person *implements* *Serializable*{

*private* *String* name, phone ,gender;

*// Default Constructor*

*public* person(){

    }

*// Argumented Constructor*

*public* person(*String* *name\_prime*, *String* *phone\_prime*, *String* *gender\_prime*){

*this*.name = name\_prime;

*this*.phone = phone\_prime;

*this*.gender = gender\_prime;

    }

*// setters*

*public* *void* setGender(*String* *gender\_prime*) {

*this*.gender = gender\_prime;

    }

*public* *void* setName(*String* *name\_prime*) {

*this*.name = name\_prime;

    }

*public* *void* setPhone(*String* *phone\_prime*) {

*this*.phone = phone\_prime;

    }

*// Getters*

*public* *String* getGender() {

        return gender;

    }

*public* *String* getName() {

        return name;

    }

*public* *String* getPhone() {

        return phone;

    }

*public* *String* toString() {

        return name + " " + phone + " " + gender + " ";

    }

}

*class* student *extends* person{

*private* *String* section;

*private* *int* GPA, semester;

*private* department d1;

*// Default Constructor*

*public* student(){

*super*();

    }

*// Argumented Constructor*

*public* student(*String* *name\_prime*, *String* *phone\_prime*, *String* *gender\_prime*, *String* *section\_prime*, *int* *GPA\_prime*, *int* *semester\_prime*, *department* *d1\_prime*){

*super*(name\_prime, phone\_prime, gender\_prime);

*this*.section = section\_prime;

*this*.GPA = GPA\_prime;

*this*.semester = semester\_prime;

*this*.d1 = d1\_prime;

    }

*// Setters*

*public* *void* setGPA(*int* *GPA\_prime*) {

*this*.GPA = GPA\_prime;

    }

*public* *void* setSection(*String* *section\_prime*) {

*this*.section = section\_prime;

    }

*public* *void* setSemester(*int* *semester\_prime*) {

*this*.semester = semester\_prime;

    }

*public* *void* setD1(*department* *d1\_prime*) {

*this*.d1 = d1\_prime;

    }

*// Getters*

*public* *int* getGPA() {

        return GPA;

    }

*public* *String* getSection() {

        return section;

    }

*public* *int* getSemester() {

        return semester;

    }

*public* department getD1() {

        return d1;

    }

*public* *String* toString(){

        return section + " " + (GPA + "") + " " + (semester + "") + " " + *super*.toString() + " " + d1.toString();

    }

}

*class* department *implements* *Serializable*{

*private* *String* name, location;

*// Default Constructor*

*public* department(){

    }

*// Argumented Constructor*

*public* department(*String* *name\_prime*, *String* *location\_prime*){

*this*.name = name\_prime;

*this*.location = location\_prime;

    }

*// Setters*

*public* *void* setLocation(*String* *location\_prime*) {

*this*.location = location\_prime;

    }

*public* *void* setName(*String* *name\_prime*) {

*this*.name = name\_prime;

    }

*// Getters*

*public* *String* getLocation() {

        return location;

    }

*public* *String* getName() {

        return name;

    }

*public* *String* toString(){

        return name + " " + location + " ";

    }

}

import *java.io.\**;

import *java.util.\**;

import *javax.swing.JOptionPane*;

*class* FileOperations <*T*>{

*// method to to write file*

*public* *void* writeToFile(*T* *object*, *String* *file\_name*){

        try{

*File* f = new File(file\_name + ".ser");

*ObjectOutputStream* oos;

            if(f.exists()){

                oos = new MyObjectOutputStream(new FileOutputStream(f, true));

            }

            else{

                oos = new ObjectOutputStream(new FileOutputStream(f));

            }

            if(object instanceof student){

                student s1 = (student) object;

                oos.writeObject(s1);

            }

            else if (object instanceof department){

                department d1 = (department) object;

                oos.writeObject(d1);

            }

            oos.close();

        } catch(*IOException* *e*){

            System.out.println("Can't write. Error!");

        }

    }

*// method to read from file*

*public* *void* readFromFile(*String* *file\_name*){

*ObjectInputStream* ois = null;

        try {

            ois = new ObjectInputStream(new FileInputStream(file\_name + ".ser"));

            while(true){

                if(file\_name.equalsIgnoreCase("department")){

                    department d1 = (department) ois.readObject();

                    System.out.println(d1.toString());

                }

                else if (file\_name.equalsIgnoreCase("student")){

                    student s1 = (student) ois.readObject();

                    System.out.println(s1.toString());

                }

            }

        } catch (*ClassNotFoundException* *e*) {

            System.out.println("Class don't exist!");

        } catch(*EOFException* *e*){

            try {

                ois.close();

            } catch (*Exception* *f*) {

                System.out.println("Error! ...");

            }

            return;

        } catch(*IOException* *e* ){

            System.out.println("Error!");

        }

    }

*// method to read from file*

*public* *void* display(*String* *file\_name*){

*ObjectInputStream* ois = null;

        try {

            ois = new ObjectInputStream(new FileInputStream(file\_name + ".ser"));

            while(true){

                if(file\_name.equalsIgnoreCase("department")){

                    department d1 = (department) ois.readObject();

                    System.out.println(d1.toString());

                }

                else if (file\_name.equalsIgnoreCase("student")){

                    student s1 = (student) ois.readObject();

                    JOptionPane.showMessageDialog(GUI.f2.getFrame(), s1.toString());

                }

            }

        } catch (*ClassNotFoundException* *e*) {

            System.out.println("Class don't exist!");

        } catch(*EOFException* *e*){

            try {

                ois.close();

            } catch (*Exception* *f*) {

                System.out.println("Error! ...");

            }

            return;

        } catch(*IOException* *e* ){

            System.out.println("Error!");

        }

    }

*// Search by name!*

*public* *void* searchByName(*String* *name*, *String* *file\_name*){

*ObjectInputStream* ois = null;

*Boolean* check = true;

        try {

            ois = new ObjectInputStream(new FileInputStream(file\_name + ".ser"));

            while(check){

                student s1 = (student) ois.readObject();

                if (s1.getName().equals(name)) {

*String* text = s1.toString();

                    JOptionPane.showMessageDialog(GUI.f4.getFrame(), text);

                    System.out.println("Found");

                    check = false;

                }

            }

        } catch (*ClassNotFoundException* *e*) {

            System.out.println("Class don't exist!");

        } catch(*EOFException* *e*){

            try {

                ois.close();

            } catch (*Exception* *f*) {

                System.out.println("Error! ...");

            }

            return;

        } catch(*IOException* *e* ){

            System.out.println("Error!");

        }

    }

*// Search by departments!*

*public* *void* searchByDepartments(*String* *department\_name*, *String* *file\_name*){

*ObjectInputStream* ois = null;

*Boolean* check = true;

        try {

            ois = new ObjectInputStream(new FileInputStream(file\_name + ".ser"));

            while(check){

                department s1 = (department) ois.readObject();

                if (s1.getName().equals(department\_name)) {

                    System.out.println("Found");

                    check = false;

                }

            }

        } catch (*ClassNotFoundException* *e*) {

            System.out.println("Class don't exist!");

        } catch(*EOFException* *e*){

            try {

                ois.close();

            } catch (*Exception* *f*) {

                System.out.println("Error! ...");

            }

            return;

        } catch(*IOException* *e* ){

            System.out.println("Error!");

        }

    }

*// method to update items*

*public* *void* update(*String* *select\_item*, *String* *update\_item*, *String* *file\_name*){

*ObjectInputStream* ois = null;

*ArrayList*<*department*> dep = new *ArrayList*<*department*>();

*ArrayList*<*student*> stu = new *ArrayList*<*student*>();

        try {

            ois = new ObjectInputStream(new FileInputStream(file\_name + ".ser"));

            while(true){

                if(file\_name.equalsIgnoreCase("department")){

                    department d1 = (department) ois.readObject();

                    dep.add(d1);

                }

                else if (file\_name.equalsIgnoreCase("student")){

                    student s1 = (student) ois.readObject();

                    stu.add(s1);

                }

            }

        } catch (*ClassNotFoundException* *e*) {

            System.out.println("Class don't exist!");

        } catch(*EOFException* *e*){

            try {

                ois.close();

            } catch (*Exception* *f*) {

                System.out.println("Error! ...");

            }

        } catch(*IOException* *e* ){

            System.out.println("Error!");

        }

        System.out.println(stu);

*// updatation*

        if(file\_name.equalsIgnoreCase("department")){

            for (*int* i = 0; i < dep.size(); i++) {

                if (dep.get(i).getName().equals(select\_item)) {

                    dep.get(i).setName(update\_item);

                }

            }

*File* myFile = new File(file\_name + ".ser");

            myFile.delete();

            for (*int* i = 0; i < stu.size(); i++) {

                writeToFile((T)dep.get(i), file\_name);

            }

        }

        else if (file\_name.equalsIgnoreCase("student")){

            for (*int* i = 0; i < stu.size(); i++) {

                if (stu.get(i).getName().equals(select\_item)) {

                    stu.get(i).setName(update\_item);

                }

            }

*File* myFile = new File(file\_name + ".ser");

            myFile.delete();

            for (*int* i = 0; i < stu.size(); i++) {

                writeToFile((T)stu.get(i), file\_name);

            }

        }

    }

*// method to update items*

*public* *void* delete(*String* *select\_item*, *String* *file\_name*){

*ObjectInputStream* ois = null;

*ArrayList*<*department*> dep = new *ArrayList*<*department*>();

*ArrayList*<*student*> stu = new *ArrayList*<*student*>();

        try {

            ois = new ObjectInputStream(new FileInputStream(file\_name + ".ser"));

            while(true){

                if(file\_name.equalsIgnoreCase("department")){

                    department d1 = (department) ois.readObject();

                    dep.add(d1);

                }

                else if (file\_name.equalsIgnoreCase("student")){

                    student s1 = (student) ois.readObject();

                    stu.add(s1);

                }

            }

        } catch (*ClassNotFoundException* *e*) {

            System.out.println("Class don't exist!");

        } catch(*EOFException* *e*){

            try {

                ois.close();

            } catch (*Exception* *f*) {

                System.out.println("Error! ...");

            }

        } catch(*IOException* *e* ){

            System.out.println("Error!");

        }

        System.out.println(stu);

*// updatation*

        if(file\_name.equalsIgnoreCase("department")){

            for (*int* i = 0; i < dep.size(); i++) {

                if (dep.get(i).getName().equalsIgnoreCase(select\_item)) {

                    dep.remove(i);

                    break;

                }

            }

*File* myFile = new File(file\_name + ".ser");

            myFile.delete();

            for (*int* i = 0; i < stu.size(); i++) {

                writeToFile((T)dep.get(i), file\_name);

            }

        }

        else if (file\_name.equalsIgnoreCase("student")){

            for (*int* i = 0; i < stu.size(); i++) {

                if (stu.get(i).getName().equalsIgnoreCase(select\_item)) {

                    stu.remove(i);

                    break;

                }

            }

*File* myFile = new File(file\_name + ".ser");

            myFile.delete();

            for (*int* i = 0; i < stu.size(); i++) {

                writeToFile((T)stu.get(i), file\_name);

            }

        }

    }

}

import *java.io.\**;

*class* MyObjectOutputStream *extends* *ObjectOutputStream*{

*// Default Constructor*

*public* MyObjectOutputStream() *throws* *IOException*{

*super*();

    }

*// Argumented Constructor*

*public* MyObjectOutputStream(*OutputStream* *o*) *throws* *IOException*{

*super*(o);

    }

*// Header so that it doesn't ruin the code*

*public* *void* writeStreamHeader(){

*// it doesn't write header*

    }

}

**Buttons Operations:**

*class* buttons *implements* *ActionListener*{

*private* *JButton* button;

*// Default Constructor*

*public* buttons(){

    }

*public* *JButton* newButton(*JFrame* *frame*, *int* *x*, *int* *y*, *int* *width*, *int* *height*, *String* *text*){

*// Creating a new button*

*this*.button = new JButton();

*this*.button.setText(text);

*this*.button.setBounds(x, y, width, height);

*this*.button.addActionListener(*this*);

*this*.button.setBackground(Color.white);

*// Imported frame as it'd add on top of the above screen*

        frame.add(button);

        return button;

    }

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

        if(e.getSource() == button && button.getX() == 300 && button.getY() == 100){

*JFrame* frame2 = GUI.f2.getFrame();

            frame2.setVisible(false);

            GUI.f3.addButtons();

            GUI.f3.setVisible(true);

*//frame.dispose();*

            System.out.println("Add ");

        }

        else if(e.getSource() == button && button.getX() == 300 && button.getY() == 300){

*JFrame* frame2 = GUI.f2.getFrame();

            frame2.setVisible(false);

            GUI.f4.addButtons();

            GUI.f4.setVisible(true);

            System.out.println("Search ");

        }

        else if(e.getSource() == button && button.getX() == 300 && button.getY() == 500){

*JFrame* frame2 = GUI.f2.getFrame();

            frame2.setVisible(false);

            GUI.f5.addButtons();

            GUI.f5.setVisible(true);

            System.out.println("Update ");

        }

        else if(e.getSource() == button && button.getX() == 550 && button.getY() == 100){

*JFrame* frame2 = GUI.f2.getFrame();

            frame2.setVisible(false);

            GUI.f6.addButtons();

            GUI.f6.setVisible(true);

            System.out.println("Delete ");

        }

        else if(e.getSource() == button && button.getX() == 550 && button.getY() == 300){

            GUI.f1.display("student");

            System.out.println("Display ");

        }

        else if(e.getSource() == button && button.getX() == 395){

            System.out.println("Clicked Me :)");

        }

        else if(e.getSource() == button && button.getX() == 605){

            System.out.println("Exit on page 1 ");

        }

        else if(e.getSource() == button && button.getX() == 150 && button.getY() == 640){

*String* name = GUI.f3.t1.getFieldText();

*String* phone = GUI.f3.t2.getFieldText();

*String* gender = GUI.f3.t3.getFieldText();

*String* deptName = GUI.f3.t4.getFieldText();

*String* deptLoc = GUI.f3.t5.getFieldText();

*String* GPA = GUI.f3.t6.getFieldText();

*String* semester = GUI.f3.t7.getFieldText();

*String* section = GUI.f3.t8.getFieldText();

            department d3 = new department(deptName, deptLoc);

            student s4 = new student(name, phone, gender, section, Integer.parseInt(GPA), Integer.parseInt(semester), d3);

            GUI.f1.writeToFile(s4, "student");

            JOptionPane.showMessageDialog(GUI.f3.getFrame(), "Added!");

            GUI.f3.setVisible(false);

            GUI.f3.getFrame().dispose();

            GUI.f2.setVisible(true);

            System.out.println("SUBMIT FIELDS");

        }

        else if(e.getSource() == button && button.getX() == 380 && button.getY() == 640){

*JFrame* frame3 = GUI.f3.getFrame();

            frame3.setVisible(false);

            GUI.f2.setVisible(true);

            System.out.println("HOME");

        }

        else if(e.getSource() == button && button.getX() == 150 && button.getY() == 150){

*String* text = GUI.f4.t9.getFieldText();

            GUI.f1.searchByName(text, "student");

            System.out.println("Search on search");

        }

        else if(e.getSource() == button && button.getX() == 151 && button.getY() == 150){

*String* oldName = GUI.f5.t10.getFieldText();

*String* newName = GUI.f5.t11.getFieldText();

            GUI.f1.update(oldName, newName, "student");

            JOptionPane.showMessageDialog(GUI.f5.getFrame(), "Updated!");

            System.out.println(oldName);

            System.out.println(newName);

            System.out.println("updated on update");

        }

        else if(e.getSource() == button && button.getX() == 152 && button.getY() == 150){

*String* name = GUI.f6.t12.getFieldText();

            GUI.f1.delete(name, "student");

            JOptionPane.showMessageDialog(GUI.f5.getFrame(), "Deleted!");

            System.out.println("updated on update");

        }

        else if(e.getSource() == button && button.getX() == 380 && button.getY() == 150){

*JFrame* frame3 = GUI.f4.getFrame();

            frame3.setVisible(false);

*JFrame* frame5 = GUI.f5.getFrame();

            frame5.setVisible(false);

*JFrame* frame6 = GUI.f6.getFrame();

            frame6.setVisible(false);

            GUI.f2.setVisible(true);

            System.out.println("HOME");

            System.out.println("HOME on Search");

        }

    }

}