Question 1

```java
import java.awt.Button;
import java.awt.FlowLayout;
import java.awt.Frame;
import java.awt.TextField;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class UpperLowerApplication extends Frame implements ActionListener {

    private static final long serialVersionUID = 1L;
    private Button button_upper = null;
    private Button button_lower = null;
    private TextField text_field = null;

    UpperLowerApplication() {
        super("Upper Lower Application");

        text_field = new TextField(10);
        this.add(text_field);

        button_upper = new Button("To Uppercase");
        this.add(button_upper);
        button_upper.addActionListener(this);

        button_lower = new Button("To Lowercase");
        this.add(button_lower);
        button_lower.addActionListener(this);

        this.setLayout(new FlowLayout());
        this.setSize(320, 75);
        this.setVisible(true);
    }

    @Override
    public void actionPerformed(ActionEvent a) {

        if (a.getSource().equals(button_lower)) {
            text_field.setText(text_field.getText().toLowerCase());
        } else {
            text_field.setText(text_field.getText().toUpperCase());
        }
    }

    public static void main(String[] args) {
        new UpperLowerApplication();
    }
}
```

Question 2

```
import java.awt.Button;
import java.awt.FlowLayout;
import java.awt.Frame;
import java.awt.Label;
import java.awt.TextField;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class MultiplicationApplication extends Frame implements ActionListener {

    private static final long serialVersionUID = 1L;
    private Button button_multiply = null;
    private TextField field_a = null;
    private TextField field_b = null;
    private TextField field_result = null;

    MultiplicationApplication() {
        super("Multiply Application");

        this.add(new Label("A:"));
        field_a = new TextField(2);
        this.add(field_a);

        this.add(new Label("B:"));
        field_b = new TextField(2);
        this.add(field_b);

        this.add(new Label("A x B:"));
        field_result = new TextField(4);
        field_result.setEditable(false);
        this.add(field_result);

        button_multiply = new Button("Calculate");
        this.add(button_multiply);
        button_multiply.addActionListener(this);

        this.setLayout(new FlowLayout());
        this.setSize(400, 75);
        this.setVisible(true);
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        double a = Double.parseDouble(field_a.getText());
        double b = Double.parseDouble(field_b.getText());
        Double result = a*b;
        field_result.setText(result.toString());
    }

    public static void main(String[] args) {
        new MultiplicationApplication();
    }
}
```

Question 3

Gender.java

```
public enum Gender {
    male,female;
}
```

Student.java

```
import java.awt.TextField;
import java.util.Map;

public class Student {

    private Programme programme = null;
    private String name = "";
    private int id_number = 0;
    private String telephone = "";
    private Gender sex = Gender.male;
    private String address = "";

    public Student(String name, int id, String telephone, Gender sex,
                                                                String address) {
        this.name = name;
        this.id_number = id;
        this.telephone = telephone;
        this.sex = sex;
        this.address = address;
    }

    public void display(Map<String, TextField> fields) {
        fields.get("Programme Title").setText(programme.getTitle());
        fields.get("Academic Year").setText(programme.getAcademic_year());
        fields.get("Calendar Year").setText(programme.getCalendar_year());
        fields.get("Number Students").setText(programme.getNumberStudents());
        fields.get("Student Name").setText(name);
        fields.get("Student Id").setText(getId_number());
        fields.get("Student Telephone").setText(telephone);
        fields.get("Student Sex").setText(getSex());
        fields.get("Student Address").setText(address);
    }

    public void setProgramme(Programme programme) {
        this.programme = programme;
    }

    public Programme getProgramme() {
        return programme;
    }

    public String getName() {
        return name;
    }

    public String getId_number() {
        return new Integer(id_number).toString();
    }

    public String getTelephone() {
        return telephone;
    }
```

```
    public String getSex() {
        return (sex == Gender.male)? "male" : "female";
    }

    public String getAddress() {
        return address;
    }
}
```

Programme.java

```
import java.util.Calendar;

public class Programme {

    private String title = "unknow";
    private int academic_year = 0;
    private int calendar_year = Calendar.getInstance().get(Calendar.YEAR);
    private Student[] students;

    public Programme(String title, int academic_year, Student[] students) {
        this.title = title;
        this.academic_year = academic_year;
        this.students = students;
        for(int i=0; i<students.length; i++) {
            students[i].setProgramme(this);
        }
    }

    public String getTitle() {
        return title;
    }

    public String getAcademic_year() {
        return new Integer(academic_year).toString();
    }

    public String getCalendar_year() {
        return new Integer(calendar_year).toString();
    }

    public Student[] getStudents() {
        return students;
    }

    public String getNumberStudents() {
        return new Integer(students.length).toString();
    }
}
```

BrowserApplication.java

```
import java.awt.Button;
import java.awt.Frame;
import java.awt.GridLayout;
import java.awt.Label;
import java.awt.TextField;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.LinkedHashMap;
import java.util.Map;
import java.util.Map.Entry;

public class BrowserApplication extends Frame implements ActionListener {

    private static final long serialVersionUID = 1L;
    Map<String, TextField> fields = new LinkedHashMap<String, TextField>();
    private Button button_next = new Button(">");
    private Button button_prev = new Button("<");
    private Programme[] programme = new Programme[2];
    private int programme_index = 0;
    private int student_index = 0;

    BrowserApplication() {
        super("Student Record Browser");

        Student[] s1 = {
              new Student("Peter", 1234, "08164895", Gender.male, "101 fake"),
              new Student("Emmy", 5678, "08974658", Gender.female, "22 street") };
        programme[0] = new Programme("ESCAO", 1, s1);

        Student[] s2 = {
               new Student("John", 1010, "0801234567", Gender.male, "unknow"),
               new Student("Leia", 1234, "0801010101", Gender.female, "big fake") };
        programme[1] = new Programme("EENG", 3, s2);

        fields.put("Programme Title", new TextField());
        fields.put("Academic Year", new TextField());
        fields.put("Calendar Year", new TextField());
        fields.put("Number Students", new TextField());
        fields.put("Student Name", new TextField());
        fields.put("Student Id", new TextField());
        fields.put("Student Telephone", new TextField());
        fields.put("Student Sex", new TextField());
        fields.put("Student Address", new TextField());

        for(Entry<String, TextField> entry : fields.entrySet()) {
            this.add(new Label(entry.getKey()));
            this.add(entry.getValue());
            entry.getValue().setEditable(false);
        }

        this.add(button_prev);
        button_prev.addActionListener(this);
        this.add(button_next);
        button_next.addActionListener(this);

        this.setLayout(new GridLayout(10,2));
        this.setSize(400,250);
        this.setResizable(false);
        this.setVisible(true);

        programme[0].getStudents()[0].display(fields);
    }
```

```
    @Override
    public void actionPerformed(ActionEvent e) {
        if(e.getSource().equals(button_prev)) {
            if(student_index > 0 ) {
                student_index--;
                programme[programme_index].getStudents()[student_index].display(fields);
            }
            else {
                if(programme_index > 0) {
                    programme_index--;
                    student_index = programme[programme_index].getStudents().length - 1;
                   programme[programme_index].getStudents([student_index].display(fields);
                }
            }
        }
        else if(e.getSource().equals(button_next)) {
            if(student_index < programme[programme_index].getStudents().length-1 ) {
                student_index++;
                programme[programme_index].getStudents()[student_index].display(fields);
            }
            else {
                if(programme_index < programme.length-1) {
                    student_index = 0;
                    programme_index++;
                    programme[programme_index].getStudents()[student_index].display(fields);
                }
            }
        }
    }


    public static void main(String[] args) {
        new BrowserApplication();
    }
}
```