Lab05

Shayne Lewis

100506658

Fxml file

```
<?import javafx.geometry.Insets?>
<?import javafx.scene.layout.GridPane?>
<?import javafx.scene.layout.*?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.text.Text?>
<?import javafx.scene.control.TableView?>
<?import javafx.scene.control.TableColumn?>

<GridPane alignment="center" hgap="10" vgap="10" xmlns:fx="http://javafx.com/fxml" fx:controller="sample.Controller">

    <TableView fx:id="tableView" GridPane.columnIndex="0" GridPane.rowIndex="0">
        <columns>
            <TableColumn fx:id="studentID" text="SID"></TableColumn>
            <TableColumn fx:id="assignments" text="Assignments"></TableColumn>
            <TableColumn fx:id="midterm" text="Midterm"></TableColumn>
            <TableColumn fx:id="finalExam" text="Final Exam"></TableColumn>
            <TableColumn fx:id="finalMark" text="Final Mark"></TableColumn>
            <TableColumn fx:id="letterGrade" text="Letter Grade"></TableColumn>
        </columns>
    </TableView>


</GridPane>
```

StudentRecord.java

```
package sample;

public class StudentRecord{
    private String studentID;
    private float assignments;
    private float midterm;
    private float finalExam;
    private float finalMark;
    private String letterGrade;

    public StudentRecord(String studentID, float assignments, float midterm, float finalExam){
        this.studentID = studentID;
        this.assignments = assignments;
        this.midterm = midterm;
        this.finalExam = finalExam;
        this.finalMark = FinalMark();
        this.letterGrade = LetterGrade();
    }

    //calc final grade
    private float FinalMark(){
        return ((0.2f)*assignments) + ((0.3f)*midterm) + ((0.5f)*finalExam);
    }
```

```java

    //letter of final grade
    private String LetterGrade(){
        String letter = "";
        if(finalMark < 50.0f){
            letter = "F";
        }
        else if(finalMark < 60.0f){
            letter = "D";
        }
        else if (finalMark < 70.0f){
            letter = "C";
        }
        else if (finalMark < 80.0f){
            letter = "B";
        }
        else if (finalMark < 100.1f){
            letter = "A";
        }
        return letter;
    }

```

```java

    //getters
    public String getStudentID(){
        return studentID;
    }

    public float getMidterm(){
        return midterm;
    }

    public float getAssignments(){
        return assignments;
    }

    public float getFinalExam(){
        return finalExam;
    }

    public float getFinalMark(){
        return finalMark;
    }

    public String getLetterGrade(){
        return letterGrade;
    }
}

```

DataSource.java

```java
package sample;

import javafx.collections.FXCollections;
import javafx.collections.ObservableList;

public class DataSource {
    public static ObservableList<StudentRecord> getAllMarks() {
        ObservableList<StudentRecord> marks = FXCollections.observableArrayList();
        // Student ID, Assignments, Midterm, Final exam
        marks.add(new StudentRecord( studentID: "100100100",  assignments: 75.0f,  midterm: 68.0f,  finalExam: 54.25f));
        marks.add(new StudentRecord( studentID: "100100101",  assignments: 70.0f,  midterm: 69.25f,  finalExam: 51.5f));
        marks.add(new StudentRecord( studentID: "100100102",  assignments: 100.0f,  midterm: 97.0f,  finalExam: 92.5f));
        marks.add(new StudentRecord( studentID: "100100103",  assignments: 90.0f,  midterm: 88.5f,  finalExam: 68.75f));
        marks.add(new StudentRecord( studentID: "100100104",  assignments: 72.25f,  midterm: 74.75f,  finalExam: 58.25f));
        marks.add(new StudentRecord( studentID: "100100105",  assignments: 85.0f,  midterm: 56.0f,  finalExam: 62.5f));
        marks.add(new StudentRecord( studentID: "100100106",  assignments: 70.0f,  midterm: 66.5f,  finalExam: 61.75f));
        marks.add(new StudentRecord( studentID: "100100107",  assignments: 55.0f,  midterm: 47.0f,  finalExam: 50.5f));
        marks.add(new StudentRecord( studentID: "100100108",  assignments: 40.0f,  midterm: 32.5f,  finalExam: 27.75f));
        marks.add(new StudentRecord( studentID: "100100109",  assignments: 82.5f,  midterm: 77.0f,  finalExam: 74.25f));

        return marks;

    }
}
```

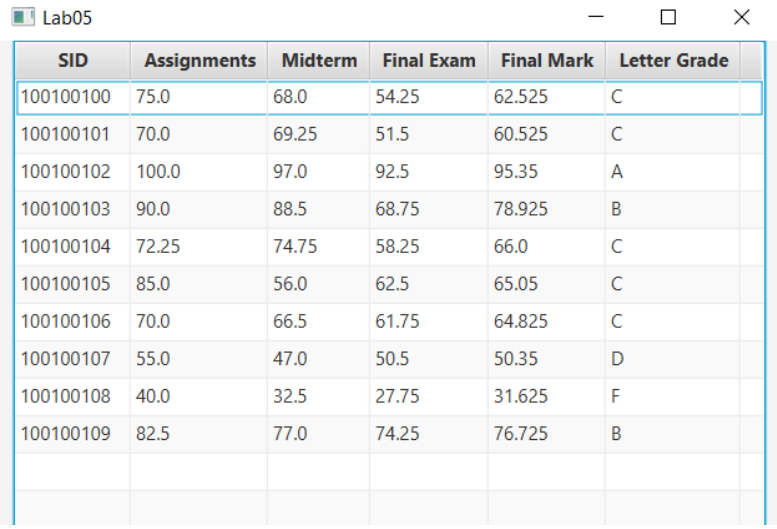
Controller.java

```java
public class Controller {
    @FXML
    private TableView tableView;
    @FXML
    private TableColumn studentID;
    @FXML
    private TableColumn midterm;
    @FXML
    private TableColumn assignments;
    @FXML
    private TableColumn finalExam;
    @FXML
    private TableColumn finalMark;
    @FXML
    private TableColumn letterGrade;

    @FXML
    public void initialize(){
        studentID.setCellValueFactory(new PropertyValueFactory<>( s: "studentID"));
        midterm.setCellValueFactory(new PropertyValueFactory<>( s: "midterm"));
        assignments.setCellValueFactory(new PropertyValueFactory<>( s: "assignments"));
        finalExam.setCellValueFactory(new PropertyValueFactory<>( s: "finalExam"));
        letterGrade.setCellValueFactory(new PropertyValueFactory<>( s: "letterGrade"));
        finalMark.setCellValueFactory(new PropertyValueFactory<>( s: "finalMark"));

        tableView.setItems(DataSource.getAllMarks());
    }
}
```

Main.java

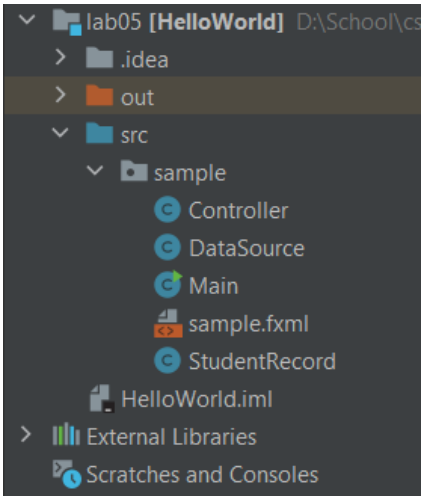
```java
package sample;

import javafx.application.Application;
import javafx.fxml.FXMLLoader;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.scene.text.Text;
import javafx.stage.Stage;

import static java.awt.Font.*;
import static javafx.scene.text.Font.font;

public class Main extends Application {

    @Override
    public void start(Stage primaryStage) throws Exception{
        Parent root = FXMLLoader.load(getClass().getResource( name: "sample.fxml"));
        primaryStage.setTitle("Lab05");
        primaryStage.setScene(new Scene(root,  v: 500,  v1: 400));
        primaryStage.show();
    }

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

Output



SID	Assignments	Midterm	Final Exam	Final Mark	Letter Grade
100100100	75.0	68.0	54.25	62.525	C
100100101	70.0	69.25	51.5	60.525	C
100100102	100.0	97.0	92.5	95.35	A
100100103	90.0	88.5	68.75	78.925	B
100100104	72.25	74.75	58.25	66.0	C
100100105	85.0	56.0	62.5	65.05	C
100100106	70.0	66.5	61.75	64.825	C
100100107	55.0	47.0	50.5	50.35	D
100100108	40.0	32.5	27.75	31.625	F
100100109	82.5	77.0	74.25	76.725	B

Local repo



Github repo: https://github.com/shayne-lewis/csci2020u_shaynelewis