Assignment #3

Objectives:

- Use inner class to implement different ActionListener
- Use Swing to create a GUI
- Use multiple frames

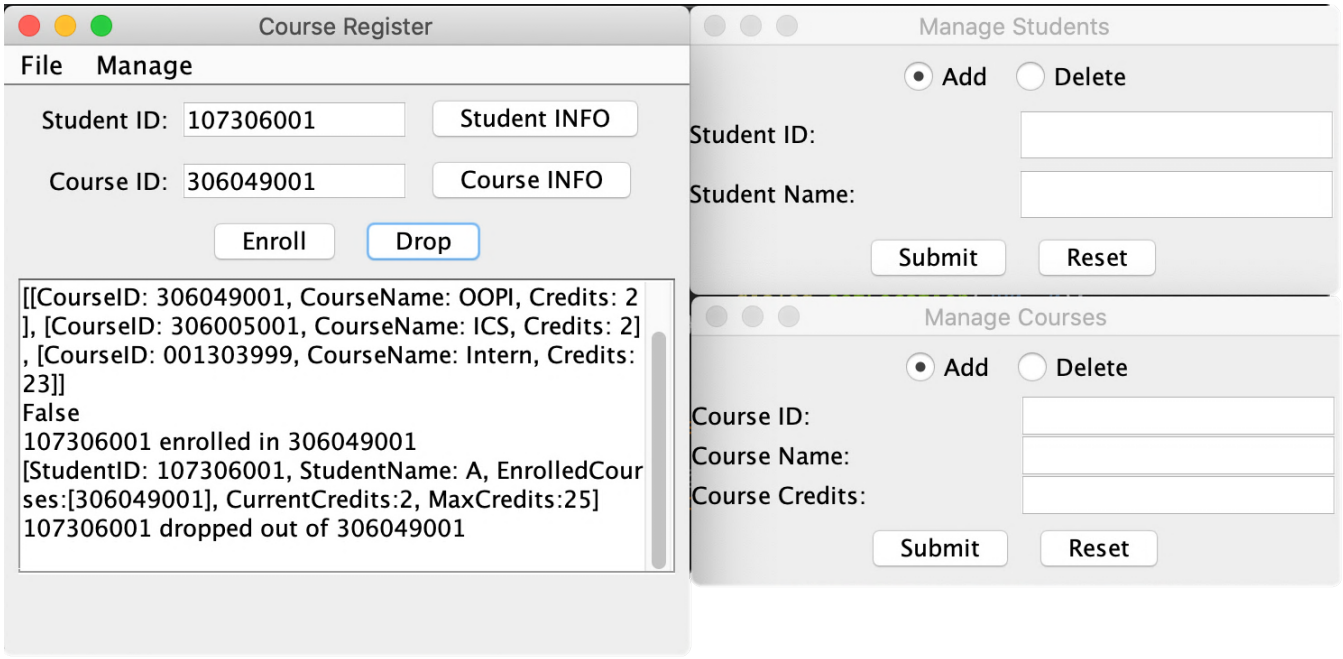


Figure 1 Sample output and GUI.

Figure 2 is the relationship of each class and detailed descriptions are not described. They are mentioned in tables on next page.

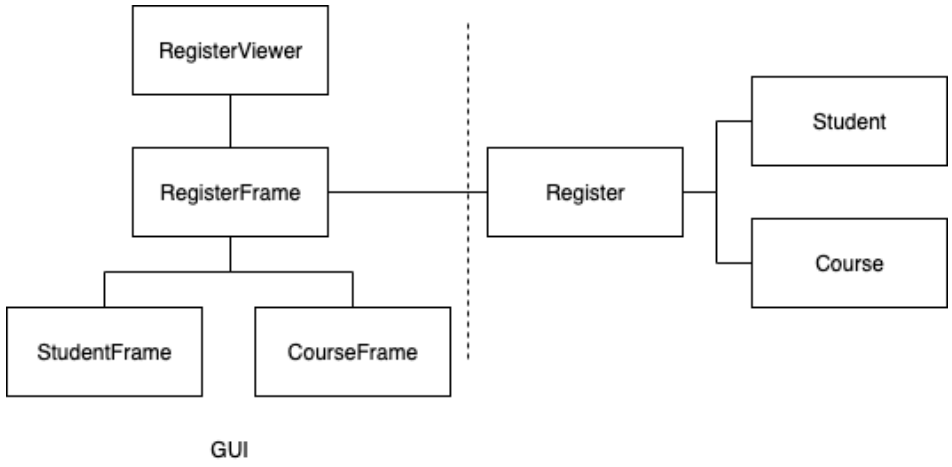


Figure 2 Relationship of each classes.

- Main method is in **RegisterViewer**.

Class description:

Student	
Modifier and type	**Method (or Variable) and description**
Instance variable	
String	studentID The student ID.
String	studentName The student's name
ArrayList<String>	enrolledCourses An ArrayList that holds all **course ID** have been selected
int	currentCredits current credits
int	maxCredits credits limit
Constructor	
Student(String studentID, String name) **Constructs a student object with given student id, name, set currentCredits as 0, set maxCredits as 25 and initialize enrolledCourses.**	
Instance methods	
-	Getter: studentID, studentName, enrolledCourses, currentCredits, maxCredits. Setter: currentCredits, maxCredits.
String	toString() Return a String description of the student. (See output in figure 1.)

Course	
Modifier and type	**Method (or Variable) and description**
Instance variable	
String	courseID The course number of this course.
String	courseName The course name of this course.
int	credits The credits of the course.
Constructor	
Course(String id, String name, int credits) **Constructs a Course object with given od, name, and credits.**	
Instance methods	
-	Getter for all attributes. No setter required.
String	toString() Return a String description of the course. (See output in figure 1.)

Register	
Modifier and type	**Method (or Variable) and description**
Instance variable	
ArrayList<Student>	studentList An ArrayList that holds all **student object**.
ArrayList<Course>	courseList An ArrayList that holds all **course object**.
Constructor	
Register() **Constructs a Register object and initialize studentList and courseList.**	
Instance methods	
-	Getter: studentList, courseList
void	addStudent(String id, String name) Create a student object by given parameters and add the student object into **studentList.**
void	addCourse(String id, String name, int credits) Create a course object by given parameters and add the course object into **courseList.**
Student	findStudent(String studentID) Find the student object in **studentList** by **studentID**. If found, returns the Student object. Otherwise, returns null.
Course	findCourse(String course) Find the student object in **courseList** by **courseID**. If found, returns the Course object. Otherwise, returns null.
boolean	enrollCourse(String studentID, String courseID) 1. Find the student object by given id and find the course object by given id. 2. If both objects can be found: A. Check if (1) the currentCredits of the student after adding the course is less than maxCredits and (2) the student hasn't enrolled in the course. B. If all conditions are met. Do C and D. C. Adjust currentCredits of the student D. Add the course's **ID** to the student's **enrolledCourse**. 3. Return false if there is any wrong operation.
boolean	dropCourse(String studentID, String courseID) 1. Find the student object by given id and find the course object by given id. 2. If both objects can be found: A. Check if the student is enrolled in the course by the courseID. B. If student is in the course, do C and D C. Adjust currentCredits of the student D. Remove the course's **ID** from the student's **enrolledCourse**. 3. Return false if there is any wrong operation.
void	removeStudent(String studentID) Remove the student object in the list by given student id.
void	removeCourse(String courseID) Remove the course object in the list by given course id.

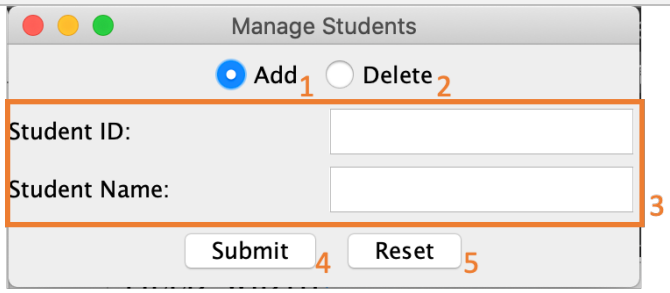
RegisterFrame Extends from JFrame	
Modifier and type	**Method (or Variable) and description**
Constant variable	
int	FRAME_WIDTH The width of the frame. 380
int	FRAME_HEIGHT The height of the frame. 360
int	FIELD_WIDTH The width of the TextField. 10
Int	AREA_WIDTH The width of the TextArea. 30
int	AREA_HEIGHT The height of the TextArea. 10
Instance variable	
Register	register
JPanel	panel
JLabel	studentIDLabel, courseIDLabel
JTextField	studentIDField, courseIDField
JButton	studentInfoButton, courseInfoButton, enrollButton, dropButton
JScrollPane	scrollPane
JTextArea	outputTextArea
JMenuBar	menuBar
Constructor and Description	
RegisterFrame() Constructs a RegisterFrame. In the constructor you have to instantiates a register object, add some student and course to this register, set the GUI title as "Course Register", and set the frame size by constant variables, FRAME_WIDTH and FREAME_HEIGHT. And then call all help methods to create a GUI. Don't forget to add the menu bar to the frame.	

`register.addCourse("306049001", "OOPI", 2);`
`register.addCourse("306005001", "ICS", 2); //Introduction to Computer Science`
`register.addCourse("001303999", "Intern", 23);`
`register.addStudent("107306001", "A");`
`register.addStudent("107306010", "B");`	

Instance methods	
void	createStudentIDComp() Instantiates a JLabel studentIDLabel, a JTextField studentIDField with FIELD_WIDTH, and a JButton studentInfoButton and define an inner class which implements ActionListener then assign it to studentInfoButton. When the button is clicked, it will perform the corresponding jobs: 1. Get the input value of studentIDField 2. Find the student object in register by given value and return it. 3. If the object isn't null, append toString() to outputTextArea. Otherwise, append "False".
void	createCourseIDComp() Instantiates a JLabel courseIDLabel, a JTextField courseIDField with FIELD_WIDTH, and a JButton courseInfoButton and define an inner class which implements ActionListener then assign it to courseInfoButton. When the button is clicked, it will perform the corresponding jobs: 1. Get the input value of courseIDField 2. Find the course object in register by given value and return it. 3. If the object isn't null, append toString() to outputTextArea. Otherwise, append "False".

<table>
<tr><td>void</td><td>createEnrollBtn()
Instantiates a JButton enrollButton and define an inner class which implements ActionListener then assign it to the button. When the button is clicked, it will perform the corresponding jobs:
1. Get the input values of studentIDField and courseIDField
2. Execute register.enrollCourse(…)
3. Use the return value from 2 and append the result to outputTextArea based on the following:
True: “studentID enrolled in courseID” / False: “False”</td></tr>
<tr><td>void</td><td>createDropBtn()
Instantiates a JButton dropButton and define an inner class which implements ActionListener then assign it to the button. When the button is clicked, it will perform the corresponding jobs:
1. Get the input values of studentIDField and courseIDField
2. Execute register.dropCourse(…)
3. Use the return value from 2 and append the result to outputTextArea based on the following:
True: “studentID dropped out of courseID” / False: “False”</td></tr>
<tr><td>void</td><td>createOutputArea()
1. Instantiates the JTextArea and JScrollPane, the add the text area to the scroll pane.
2. Sets the line-wrapping policy of the text area. If the lines are too long to fit within the allocated with, they will be wrapped.</td></tr>
<tr><td>void</td><td>createMenuBar()
<table><tr><th>JMenu</th><th>File</th><th>Manage</th></tr><tr><td rowspan="3">JMenuItem</td><td>Show students</td><td>Student</td></tr><tr><td>Show courses</td><td rowspan="2">Course</td></tr><tr><td>Exit</td></tr></table>Instantiates all components in the menu bar in this method or by using help methods.
1. When “Show students” is clicked, it will perform the corresponding jobs:
A. Call getter of studentList in Register.
B. Append the returned list as String to outputTextArea.
2. When “Show courses” is clicked, it will perform the corresponding jobs:
A. Call getter of courseList in Register.
B. Append the returned list as String to outputTextArea.
3. When “Student” is clicked, it will perform the corresponding jobs:
A. Instantiate a StudentFrame object which title is “Manage Students” and display it next to “Course Register”. Hint: JFrame.setLocation(x, y)
4. When “Course” is clicked, it will perform the corresponding jobs:
A. Instantiate a CourseFrame object which title is “Manage Courses” and display it next to “Course Register”. Hint: JFrame.setLocation(x, y)
5. When “Exit” is clicked, end the program.</td></tr>
<tr><td>void</td><td>createPanel()
Instantiates a JPanel and add all components into it, then add the panel to the frame.</td></tr>
<tr><td>-</td><td>Getter: register</td></tr>
</table>

StudentFrame Extends from JFrame	
Modifier and type	**Method (or Variable) and description**
Constant variable	
int	FRAME_WIDTH The width of the frame. 360
int	FRAME_HEIGHT The height of the frame. 160
int	FIELD_WIDTH The width of the TextField. 10
Instance variable	
JRadioButton	addButton, deleteButton
JLabel	studentIDLabel, studentNameField
JTextField	studentIDField, studentNameField
JButton	submitButton, resetButton
Constructor and Description	
StudentFrame() Constructs a StudentFrame. Set the frame size by constant variables, FRAME_WIDTH and FREAME_HEIGHT. And then call all help methods to create a GUI.	
GUI instructions	



Please use help methods to implement this frame.

Programming instructions:

A. User can only either select "Add" or "Delete" at one time.
B. The default selected radio button is "Add".
C. When "Add" is selected, all text field are editable.
D. When "Delete" is selected, the text field for student name is not editable.
E. When "Submit" is clicked, identify which radio button (add or delete) is selected then perform the following josbs:
   1. If "Add" is selected, get input values from text fields then add a student object to studentList.
   2. If "Delete" is selected, get the input value of student id then remove the student object from studentList.
F. When "Reset" is clicked, it will empty all textfields.

Hint:

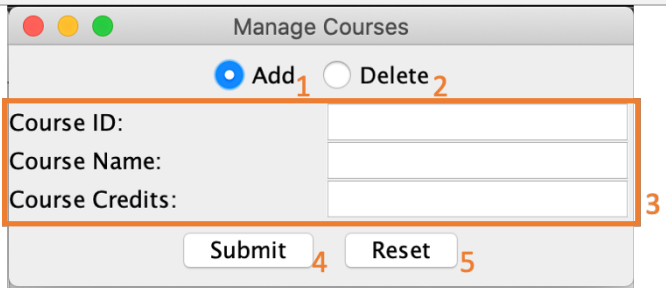
A. You can define two methods as enableFields() and disableFields() to control the text field is editable or not.
B. Call setSelected(boolean) for programming instruction B.
C. Use isSelected() to identify which radio button is selected for programming instruction E.

Layout instructions:

3 panel is required:

A. A panel which contain component 1 and 2, and is at the top of the container.
B. A panel (component 3) which applied gird layout (2x2), and is at the center of the container.
C. A panel which contain component 4 and 5, and is at the bottom of the container.

CourseFrame Extends from JFrame	
Modifier and type	**Method (or Variable) and description**
Constant variable	
int	FRAME_WIDTH The width of the frame. 360
int	FRAME_HEIGHT The height of the frame. 160
int	FIELD_WIDTH The width of the TextField. 10
Instance variable	
JRadioButton	addButton, deleteButton
JLabel	courseIDLabel, courseNameLabel, courseCreditsLabel
JTextField	courseIDField, courseNameField, courseCreditsField
JButton	submitButton, resetButton
Constructor and Description	
CourseFrame() Constructs a CourseFrame. Set the frame size by constant variables, FRAME_WIDTH and FREAME_HEIGHT. And then call all help methods to create a GUI.	
GUI instructions	



Please use help methods to implement this frame.

Programming instructions:

G. User can only either select "Add" or "Delete" at one time.
H. The default selected radio button is "Add".
I. When "Add" is selected, all text fields are editable.
J. When "Delete" is selected, the text field for course name and credits are not editable.
K. When "Submit" is clicked, identify which radio button (add or delete) is selected then perform the following josbs:
   1. If "Add" is selected, get input values from text fields then add a course object to courseList.
   2. If "Delete" is selected, get the input value of course id then remove the course object from courseList.
L. When "Reset" is clicked, it will empty all text fields.

Hint:

D. You can define two methods as enableFields() and disableFields() to control the text field is editable or not.
E. Call setSelected(boolean) for programming instruction B.
F. Use isSelected() to identify which radio button is selected for programming instruction E.

Layout instructions:

3 panel is required:

D. A panel which contain component 1 and 2, and is at the top of the container.
E. A panel (component 3) which applied gird layout (3x2), and is at the center of the container.
F. A panel which contain component 4 and 5, and is at the bottom of the container.

```
//Please add the following method in StudentFrame and CourseFrame.
private Register getRegisterFromRegisterFrame() {
	for(Frame frame: JFrame.getFrames()) {
		if(frame.getTitle().equals("Course Register")) {
			RegisterFrame registerFrame = (RegisterFrame) frame;
			return registerFrame.getRegister();
		}
	}
	return null;
}
```

RegisterViewer
main(String args[]) Use the RegisterFrame to create a GUI, then set close operation by JFrame.EXIT_ON_CLOSE and make the GUI visible.

Submission instruction:

1. Export your assignment as an executable JAR file.
2. Upload you the JAR file and the source code as ZIP file to WM5. (Two files in total.)
3. Peer evaluation for the GUI will be announced on 4/27.28.

Teammate evaluation: If you work in pair, please fulfill this form: https://forms.gle/2RRDSgmoRHvmAFmb6

Reminder: Please zip the whole project. Each team submits your work by one. Your project and file name are supposed to be like “66_HW3”, 66 is the team number.

Deadline: 4/26 23:59 (for both Mon56 and Tue23)