Assignment #2

Objectives:

- Use inner class to implement different ActionListener
- Use JButton, JLabel, JTextField, JPanel, and JFrame to create a GUI
- Read inputs from JTextField so as to manipulate objects

In chapter 8 and 17, you have basic understandings of event, listener, and GUI. Figure 1 is the screenshot of the completed output and GUI.

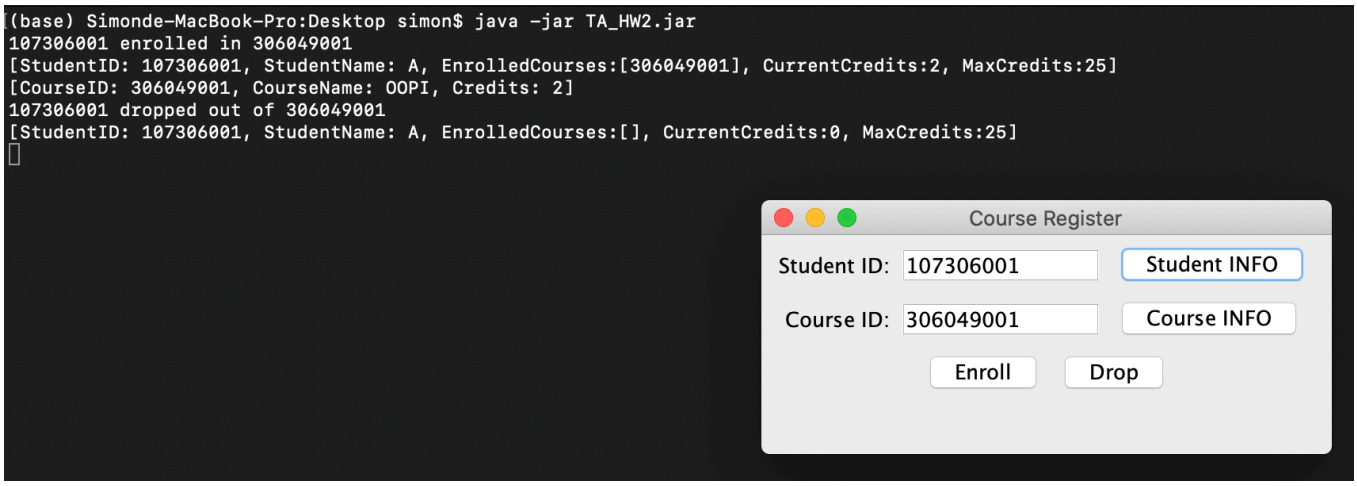


Figure 1 Sample output and GUI.

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

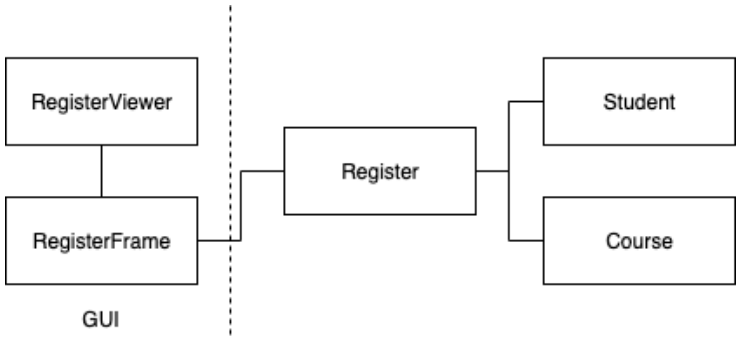


Figure 2 Relationship of each classes.

- Main method is in **RegisterViewer**.
- Read slides carefully, you would find many helpful information.

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(int 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(int 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	
-	No getter and setter for attributes are required.
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.

Tester for Register, Student, and Course:

```
public class RegisterTester {

  public static void main(String[] args) {
    Student student = new Student("1234", "A");
    System.out.println(student.toString());

    Course course = new Course("12345", "CC", 3);
    System.out.println(course.toString());

    Register register = new Register();
    register.addStudent("12345AS", "A");
    register.addCourse("99ss", "ABC", 4);

    System.out.println(register.findCourse("12345AS"));
    System.out.println(register.findStudent("99ss"));
    System.out.println(register.findStudent("12345AS"));
    System.out.println(register.findCourse("99ss"));
    System.out.println(register.enrollCourse("12345AS", "99ss")); //true
    System.out.println(register.enrollCourse("12345AS", "99ss")); //false
    System.out.println(register.enrollCourse("12345AS", "99ss123")); //false
    System.out.println(register.enrollCourse("12345AS222", "99ss")); //false
    System.out.println(register.dropCourse("12345AS", "99ss")); //true
    System.out.println(register.dropCourse("12345AS", "99ss")); //false
    System.out.println(register.dropCourse("12345AS", "99ss123")); //false
    System.out.println(register.dropCourse("12345AS222", "99ss")); //false
  }

}
```

```
Output:
[StudentID: 1234, StudentName: A, EnrolledCourses:[], CurrentCredits:0, MaxCredits:25]
[CourseID: 12345, CourseName: CC, Credits: 3]
null
null
[StudentID: 12345AS, StudentName: A, EnrolledCourses:[], CurrentCredits:0, MaxCredits:25]
[CourseID: 99ss, CourseName: ABC, Credits: 4]
true
false
false
false
true
false
false
false
```

RegisterFrame 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	
final Register	register
JPanel	panel
JLabel	studentIDLabel, courseIDLabel
JTextField	studentIDField, courseIDField
JButton	studentInfoButton, courseInfoButton, enrollButton, dropButton
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.	

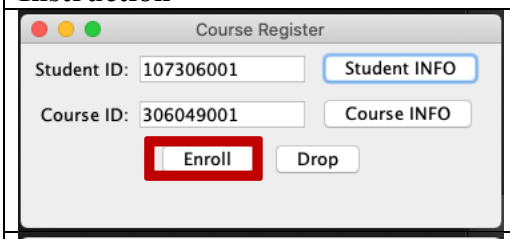
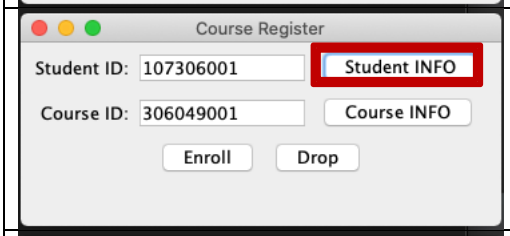
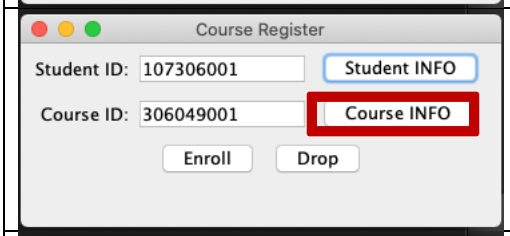
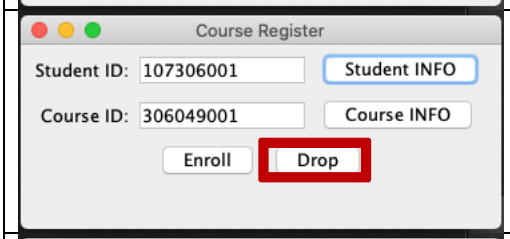
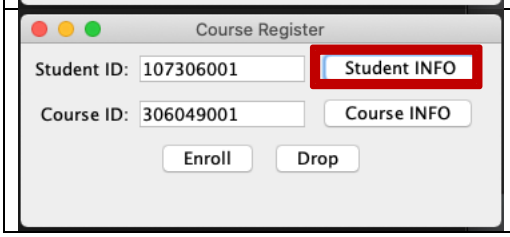
`register.addCourse("306049001", "OOPI", 2);`
`register.addCourse("306005001", "ICS", 2); //Introduction to Computer Science`
`register.addCourse("001303999", "Intern", 23);`
`register.addStudent("107306001", "A");`
`register.addStudent("107306010", "B");`	

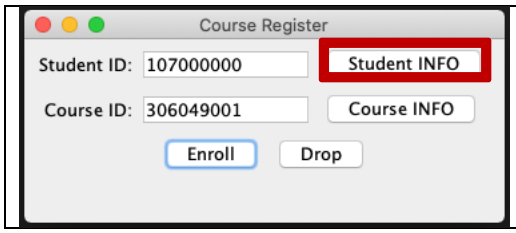
Instance methods	
void	createStudentIDComp() Instantiates a JLabel studentIDLabel, a JTextField studentIDField with FIELD_WIDTH, and a JButton studentInfoButton and define an inner class which implements ActionListener then assign it to studentInfoButton. When the button is clicked, it will perform the 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, print toString() in the console. Otherwise, print "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, print toString() in the console. Otherwise, print "False".
void	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 print the result in the console based on the following: True: "studentID enrolled in courseID" / False: "False"

void	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 print the result in the console based on the following: True: "studentID dropped out of courseID" / False: "False"
void	createPanel() Instantiates a JPanel and add all components into it, then add the panel to the frame.

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.

GUI testing instructions:

Instruction	Console Output
	107306001 enrolled in 306049001
	[StudentID: 107306001, StudentName: A, EnrolledCourses:[306049001], CurrentCredits:2, MaxCredits:25]
	[CourseID: 306049001, CourseName: OOPI, Credits: 2]
	107306001 dropped out of 306049001
	[StudentID: 107306001, StudentName: A, EnrolledCourses:[], CurrentCredits:0, MaxCredits:25]

	`False`

Submission instruction:

1. (Optional) Test **Student**, **Course**, and **Register** class on the testing website.
2. Export your assignment as an executable JAR file.
3. Upload you the JAR file and the source code as ZIP file to WM5. (Two files in total.)
4. Peer evaluation for the GUI will be announced on 4/13.14.

Teammate evaluation: If you work in pair, please fulfill this form: https://forms.gle/D3Qe4oMvU9vSTpxS7

Reminder: Please zip the whole project. Each team submits your work by one. Your project and file name are supposed to be like "66_HW2", 66 is the team number.

Deadline: 4/12 23:59 (for both Mon56 and Tue23)