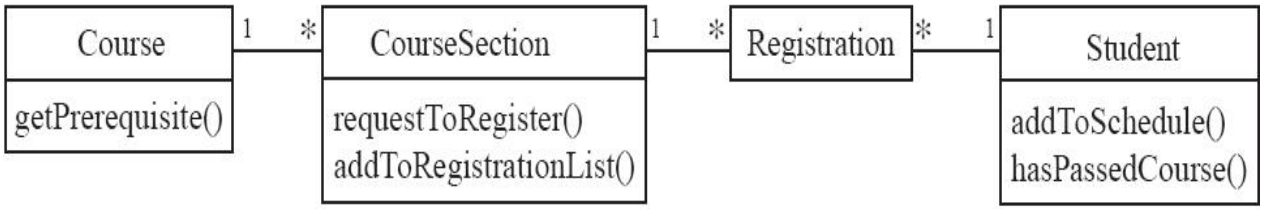
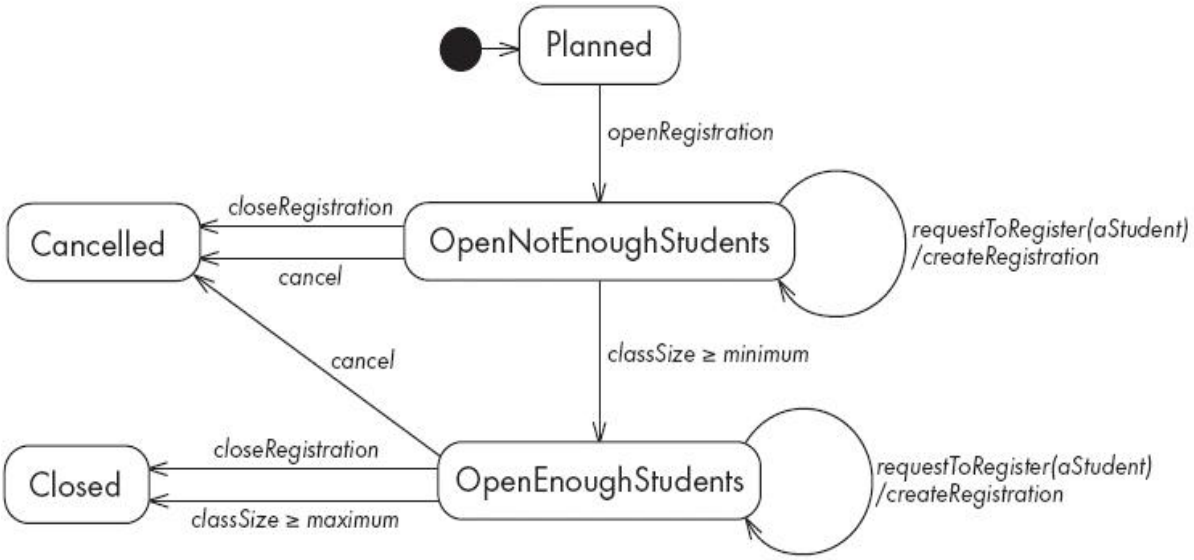
Software Engineering

Exercise Implement the class `CourseSection` based on the following (class and STD) diagrams:





(Source: T. Lethrbridge, R. Laganiere. Object oriented software engineering, 2005, section 8)

The two associations can be implemented as follows:

```
private Course course;           //  A many-to-one relation with Course
private List registrationList;   // A one-to-many relation with Registration
```

The state of a `CourseSection` object can be represented using two Boolean (instance) variables: `open` and `closedOrCanceled`. The states can be encoded as follows:

- 'Planned': `closedOrCanceled == false && open == false`
- 'Open': `open == true`
- 'NotEnoughStudents' (substate of 'Open'):
  `open == true && registrationList.size() < course.getMinimum()`
- 'EnoughStudents' (substate of 'Open'):
  `open == true && registrationList.size() >= course.getMinimum()`
- 'Canceled':
  `closedOrCanceled == true && registrationList.size() == 0`
- 'Closed' (substate course section is too full, or being taught):
  `closedOrCanceled == true && registrationList.size() > 0`

The variable `classSize` can be implemented as a call `registrationList.size()`.

Hint: You can design methods `openRegistration()`, `closeRegistration()`, `cancel()` and `requestToRegister()` to implement the events of the same name.