



## Course

here we define what a course object is.

### Variables:

```
String name          int sectionNum
String id            String courseLocation
int maxStudentsNum
int numRegisteredStudents String instructor
ArrayList<Student> students
```

### Constructors:

```
Public Course()
```

```
public Course(name, id, maxStudentsNum,
               numRegisteredStudents, studentsRegistered,
               instructor, sectionNum, courseLocation)
```

### Methods:

```
getName()          getInstructor()
setName()          setInstructor()
getId()            getSectionNum()
setId()            setSectionNum()
getMaxStudentsNum() getCourseLocation()
setMaxStudentsNum() setCourseLocation()
getNumRegisteredStudents() displayCourseInfo()
setNumRegisteredStudents() displayAllCourses()
getStudentsRegistered() displayCoursesForStudent()
setStudentsRegistered()
```

## CourseDirectory

This class is where we store an ArrayList of courses as well as an ArrayList of Students. Both an Admin and a Student have their own CourseDirectory.

### Variables:

```
# ArrayList<Course> coursesList
# ArrayList<Student> studentsList
```

```
Constructor: public CourseDirectory()
```

### Methods:

```
getStudentsList()    viewCourses()
setStudentsList()    viewNotFullCourses()
getCoursesList()     register()
createNewCourse()    withdraw()
deleteCourse()       viewRegisteredCourses()
editCourse()         setCourses()
displayCourseInfo()
registerStudent()
viewAllCourses()
viewFullCourses()
writeFullCourses()
viewStudentsInSpecificCourse()
viewAllStudentsCourses()
sortCourses()
```

## CourseRegistrationApplication

This class has the main() method.

### main()

- 1) create an Admin & store their courseDirectory w/ info from the CSV file (here we read from the csv file)
  - 2) tell the user to select if they are an Admin or Student (1st time=no students).
  - 3) If the user is an Admin, ask for username & password & display the 2 menus where they decide what they want to do. Keep getting Admin's input until they exit.
  - 4) If the user is a student, ask for username & password & display the menu of what a student can choose to do. Continue until Student exits.
  - 5) go back to the menu asking if the user is a Student or Admin & if they select "exit", then stop the program entirely
- whenever an Admin or a Student is done, serialize their courseDirectory
  - whenever an Admin or Student logs in, deserialize the file containing courseDirectory info

### Methods:

```
serializeCourseDirectory()
deserializeCourseDirectory()
coursesManagementAdminMenu()
reportsAdminMenu()
StudentMenu()
readCourseData()
    ↳ reads from csv file
```