

Project #1

For this project, you will be implementing a simplified version of the JSU student registration system. Attached is a data file, in *tab-delimited CSV format*, containing the data for the Mathematics and Computer Science courses offered by JSU in the Spring 2019 semester.

(If you wish to view this file, be sure to open it in a text editor only; DO NOT double-click the file to open it in Microsoft Excel, or else you run the risk of overwriting the file with an incorrectly-formatted version created by Excel! If you are getting errors in your program which seem to be related to the contents of the input file, such as exceptions when parsing numbers, try replacing your copy of the data file with a fresh copy from Canvas.)

Your task for this project is to implement a system which will allow the user to:

- Search for specific courses in the database by course number, which should return a list of the section(s) available for each course. *(Remember that the course number includes the subject and the number, separated by a space; for example, "CS 201". You may prompt for these values separately, or as a single string, as you wish.)*
- Register for a specific course and section by entering the corresponding CRN number. The system should not allow the user to register for classes which conflict with each other (that is, classes that are held on overlapping days and times).
- View the "trial schedule," in the form of a list of the courses that have been added to the schedule so far. Each of these options should be chosen from a simple menu, presented to the user at the console.

Here are a few tips to get you started:

• To read the data from the file, refer back to the code given in class. In this project, the input file is a *tab-delimited CSV file*. Use the code from class as a guide and remember that `'\t'` is the escape sequence for the tab delimiter (you will need this to split the lines into individual values).

Within the data file, each line corresponds to the data about one course. For traditional (lecture) courses, the data fields within each line are, in order: **course name**, **CRN number**, **course number**, **section number**, **credit hours**, **starting time** (in HH:MM format), **stopping time** (in HH:MM format), **scheduled meeting days**, **scheduled meeting room/location**, **course type**, and **instructor name(s)**.

Notice that online courses omit several of these fields and will need to be read differently (see below). To store the course data in memory as an iterable collection, it will be necessary to implement a series of simple *container classes* for data about specific courses. A **Course** class, for example, would contain instance fields for the information about a specific course: the course name and number, the CRN number, etc. You can create one object for each course from the data file, then store these objects into an iterable collection such as an **ArrayList**.

Notice, however, that the data file includes information about different *types* of courses. For example, traditional lecture-based courses have a scheduled meeting day and time and an assigned room number, while online courses do not. In this project, we will need different subclasses to represent different kinds of courses.

So, you will need to implement **Course** as an *abstract parent class*, containing information common to *all* types of courses, and *subclasses* for specific types of courses (for example, **TraditionalCourse** and **OnlineCourse**), containing any information specific to these types of courses. This will allow you to unify all of your course objects into a single collection.

Both subclasses should implement the "**toString()**" method to print the course data to the screen in a way that is specific to each type of course (see the examples below for one suggestion). The superclass **Course** should also include an abstract "**conflictsWith()**" method which accepts another **Course** object as an argument and returns a **boolean** value. This method should return **true** if the two courses conflict with each other, and **false** if they do not. Again, implement this method in both subclasses in a way that is appropriate for both types of courses.

Remember that your system must prevent registering for conflicting courses! To make this easier, use the libraries provided by Java for working with dates and times. Recall that we saw the **LocalDate** class in several of our early object-oriented programming examples to represent dates. I would recommend using **LocalTime** objects to represent the starting and stopping times for each course; see the official documentation for this class online. **LocalTime** includes such useful methods as **isBefore()** and **isAfter()**, which you may find useful for detecting overlaps.

Here is a suggested example of what the user interface should look like (the user's input is shown in bold). **NOTE: THE OUTPUT SHOWN BELOW IS ONLY AN EXAMPLE; THE CRN NUMBERS SHOWN HERE WILL DIFFER FROM THOSE IN YOUR DATABASE!**

```
1) Search Courses
2) Register for Course
3) View Trial Schedule
4) Quit
```

```
Your Choice? 1
Enter course number in the format SSNNN (for example, CS201): CS230
#21733: CS 230-001 (Fundamentals of Computing), Eric Adelaide Gamess, Lecture, 10:00 - 11:00,
MWF, Ayers Hall 257
#21734: CS 230-002 (Fundamentals of Computing), Sajib Datta, Lecture, 12:30 - 13:30, MWF,
Ayers Hall 257
#21735: CS 230-003 (Fundamentals of Computing), Nathan Lee Mayberry, Lecture, 09:15 - 10:45, TR,
Ayers Hall 363
1) Search Courses
2) Register for Course
3) View Trial Schedule
4) Quit
```

```
Your Choice? 2
Enter CRN number: 21735
Course added successfully!
1) Search Courses
2) Register for Course
3) View Trial Schedule
4) Quit
```

```
Your Choice? 1
Enter course number in the format SSNNN (for example, CS201): MS100
#21929: MS 100-001 (Intermediate Algebra for Precalculus), Janice Lynn Case, Lecture, 12:30
- 13:30, MWF, Ayers Hall 218

#21930: MS 100-002 (Intermediate Algebra for Precalculus), Janice Lynn Case, Lecture, 09:15
- 10:45, TR, Ayers Hall 218
```

#22199: MS 100-100 (Intermediate Algebra for Precalculus), Curtis L Gladen, Lecture, 08:45 - 11:00, MTWF, Houston Cole Library B03

- 1) Search Courses
- 2) Register for Course
- 3) View Trial Schedule
- 4) Quit

Your Choice? 2

Enter CRN number: 21930

ERROR: Either the course was not found, or it conflicts with your current schedule!

- 1) Search Courses
- 2) Register for Course
- 3) View Trial Schedule
- 4) Quit

Your Choice? 2

Enter CRN number: 21929

Course added successfully!

- 1) Search Courses
- 2) Register for Course
- 3) View Trial Schedule
- 4) Quit

Your Choice? 3

#21735: CS 230-003 (Fundamentals of Computing), Nathan Lee Mayberry, Lecture, 09:15 - 10:45,

TR, Ayers Hall 363

#21929: MS 100-001 (Intermediate Algebra for Precalculus), Janice Lynn Case, Lecture, 12:30

- 13:30, MWF, Ayers Hall 218

- 1) Search Courses
- 2) Register for Course
- 3) View Trial Schedule
- 4) Quit

Your Choice? 1

Enter course number in the format SSNNN (for example, CS201): CS201

#21720: CS 201-001 (Introduction to Information Technology), Nathan Lee Mayberry, Lecture, 08:45 - 09:45, MWF, Ayers Hall 357

#21721: CS 201-002 (Introduction to Information Technology), Cynthia Gunter Jensen, Lecture, 10:00 - 11:00, MWF, Ayers Hall 357

#21722: CS 201-003 (Introduction to Information Technology), Cynthia Gunter Jensen, Lecture, 11:15 - 12:15, MWF, Ayers Hall 357

#21723: CS 201-004 (Introduction to Information Technology), Cynthia Gunter Jensen, Lecture, 12:30 - 13:30, MWF, Ayers Hall 357

#21724: CS 201-005 (Introduction to Information Technology), Cynthia Gunter Jensen, Lecture, 13:45 - 14:45, MWF, Ayers Hall 357

#21725: CS 201-006 (Introduction to Information Technology), Nathan Lee Mayberry, Lecture, 09:15 - 10:45, TR, Ayers Hall 357

#21726: CS 201-007 (Introduction to Information Technology), TBA, Lecture, 11:00 - 12:30,

TR, Ayers Hall 357

#21727: CS 201-008 (Introduction to Information Technology), Matthew R Burns, Lecture, 11:00

- 12:30, TR, Ayers Hall 359

#21728: CS 201-009 (Introduction to Information Technology), Matthew R Burns, Lecture, 12:45

- 14:15, TR, Ayers Hall 357

#21729: CS 201-010 (Introduction to Information Technology), Matthew R Burns, Lecture, 14:30

- 16:00, TR, Ayers Hall 357

#21730: CS 201-011 (Introduction to Information Technology), Thomas D White, World Wide Web
#21731: CS 201-012 (Introduction to Information Technology), Cynthia Gunter Jensen, World Wide Web

- 1) Search Courses
- 2) Register for Course
- 3) View Trial Schedule
- 4) Quit

Your Choice? 2

Enter CRN number: 21731

Course added successfully!

- 1) Search Courses
- 2) Register for Course
- 3) View Trial Schedule
- 4) Quit

Your Choice? 3

#21735: CS 230-003 (Fundamentals of Computing), Nathan Lee Mayberry, Lecture, 09:15 - 10:45,

TR, Ayers Hall 363

#21929: MS 100-001 (Intermediate Algebra for Precalculus), Janice Lynn Case, Lecture, 12:30

- 13:30, MWF, Ayers Hall 218

#21731: CS 201-012 (Introduction to Information Technology), Cynthia Gunter Jensen, World Wide Web

- 1) Search Courses
- 2) Register for Course
- 3) View Trial Schedule
- 4) Quit

Your Choice? 4

Thanks for using the Student Registration System!