ICE – Chapter 8

http://birt.actuate.com/images/full_images/cfa/icons/course-catalog.png

Objectives:

- Practice storing objects in arrays
- Practice passing objects as arguments
- Practice using static variables

Assignment:

In this assignment, you are going to create a program that allows teachers to access and/or update course information for the Computer Science Department. The program will continue until the user chooses to quit.

There will only be two classes:

- Introduction to Programming, CSCI 2000, 60536, 35 seats
- Intermediate Programming, CSCI 2003, 63972, 35 seats

The user will have 5 options:

- View information about the currently available courses
  - Displays course name, course code, section number, seats available
- View all the students currently enrolled in a course
  - Choose available choice to view
  - Display all students’ ID and name
- Add a student to a course
  - Choose course to add student to
  - Enter student’s name and ID
  - Create a student object
  - Add student to correct course
- View the total enrollment for the Computer Science Department
  - This is the total number of students in ALL the courses
- Quit the program

This program will have 3 classes:

Student Class – represents a student:

- 2 instance variables:
  - name of student
  - student ID
- Argument and no-argument constructors
- Getters and setters for each instance variable.

Course Class – represents a course:

- 5 instance variables:
  - Course name
  - Course code
  - Section number
  - Available seats
  - Array to hold Student objects
- 1 static instance variable:
  - Stores total enrollment for all courses

- Argument and no-argument constructors
- Getters and setters for each instance variable, except the static one

- **addStudent method:**
  - 1 parameter
  - Adds a new student object to array
  - Update seats and total enrollment
  - If no seats are available, displays message saying the class is full.
  - No return value

- **displayCourseInfo method:**
  - No parameters
  - Displays all the course information
  - No return value

- **displayStudents method:**
  - No parameters
  - Displays all the students in a specific course
  - If no students are in the course, displays message saying no students are enrolled
  - No return value

CSCIDepartment Class – uses Student and Course classes

- Creates the two Course objects
- Creates Student object as needed
- Displays and allows user to choose from the 5 options
- Continues until the user enters 5 to quit

Programming Notes:

- You **MUST** store the Student objects in the Course object array.
- The total enrollment instance variable **MUST** be static.

Sample Execution 1:

```
Course Information Access
=========================

Options
=======
1. View Available Course Information
2. View Students Enrolled in Course
3. Add Student to a Course
4. View Total Enrollment for Department
5. Quit

Enter choice (1 - 5): 1

----------------------------------------------------

Available Courses
=================

Name:    Introduction to Programming
Course:  CSCI 2000
Section: 60536
Seats:   35

Name:    Intermediate Programming
Course:  CSCI 2003
Section: 63972
Seats:   35

----------------------------------------------------

Options
=======
1. View Available Course Information
2. View Students Enrolled in Course
3. Add Student to a Course
4. View Total Enrollment for Department
5. Quit

Enter choice (1 - 5): 2

----------------------------------------------------

Course Display Options
========================
1. Introduction to Programming
2. Intermediate Programming

Enter choice (1 or 2): 1


CSCI 2000 - 60536: Introduction to Programming

No students enrolled!
```

```
----------------------------------------------------

Options
=======
1. View Available Course Information
2. View Students Enrolled in Course
3. Add Student to a Course
4. View Total Enrollment for Department
5. Quit

Enter choice (1 - 5): 3

----------------------------------------------------

Course Options
==============
1. Introduction to Programming
2. Intermediate Programming

Enter choice (1 or 2): 2

Add a Student
=============

Enter Student's Name: Tyler Greer
Enter Student's ID: 12345678

----------------------------------------------------

Options
=======
1. View Available Course Information
2. View Students Enrolled in Course
3. Add Student to a Course
4. View Total Enrollment for Department
5. Quit

Enter choice (1 - 5): 2

----------------------------------------------------

Course Display Options
========================
1. Introduction to Programming
2. Intermediate Programming

Enter choice (1 or 2): 2


CSCI 2003 - 63972: Intermediate Programming

ID          Name
--          ----
12345678    Tyler Greer
```

```
----------------------------------------------------

Options
=======
1. View Available Course Information
2. View Students Enrolled in Course
3. Add Student to a Course
4. View Total Enrollment for Department
5. Quit

Enter choice (1 - 5): 1

----------------------------------------------------

Available Courses
=================

Name:    Introduction to Programming
Course:  CSCI 2000
Section: 60536
Seats:   35

Name:    Intermediate Programming
Course:  CSCI 2003
Section: 63972
Seats:   34

----------------------------------------------------

Options
=======
1. View Available Course Information
2. View Students Enrolled in Course
3. Add Student to a Course
4. View Total Enrollment for Department
5. Quit

Enter choice (1 - 5): 3

----------------------------------------------------

Course Options
==============
1. Introduction to Programming
2. Intermediate Programming

Enter choice (1 or 2): 1

Add a Student
=============

Enter Student's Name: Zoe Wilson
Enter Student's ID: 87654321

----------------------------------------------------
```

```
Options
=======
1. View Available Course Information
2. View Students Enrolled in Course
3. Add Student to a Course
4. View Total Enrollment for Department
5. Quit

Enter choice (1 - 5): 2

----------------------------------------------------

Course Display Options
========================
1. Introduction to Programming
2. Intermediate Programming

Enter choice (1 or 2): 1


CSCI 2000 - 60536: Introduction to Programming

ID          Name
--          ----
87654321    Zoe Wilson

----------------------------------------------------

Options
=======
1. View Available Course Information
2. View Students Enrolled in Course
3. Add Student to a Course
4. View Total Enrollment for Department
5. Quit

Enter choice (1 - 5): 1

----------------------------------------------------

Available Courses
=================

Name:    Introduction to Programming
Course:  CSCI 2000
Section: 60536
Seats:   34

Name:    Intermediate Programming
Course:  CSCI 2003
Section: 63972
Seats:   34

----------------------------------------------------
```

```
Options
=======
1. View Available Course Information
2. View Students Enrolled in Course
3. Add Student to a Course
4. View Total Enrollment for Department
5. Quit

Enter choice (1 - 5): 3

----------------------------------------------------

Course Options
==============
1. Introduction to Programming
2. Intermediate Programming

Enter choice (1 or 2): 2

Add a Student
=============

Enter Student's Name: Ryan Relinski
Enter Student's ID: 23456789

----------------------------------------------------

Options
=======
1. View Available Course Information
2. View Students Enrolled in Course
3. Add Student to a Course
4. View Total Enrollment for Department
5. Quit

Enter choice (1 - 5): 2

----------------------------------------------------

Course Display Options
========================
1. Introduction to Programming
2. Intermediate Programming

Enter choice (1 or 2): 2


CSCI 2003 - 63972: Intermediate Programming

ID          Name
--          ----
12345678    Tyler Greer
23456789    Ryan Relinski
```

```
----------------------------------------------------

Options
=======
1. View Available Course Information
2. View Students Enrolled in Course
3. Add Student to a Course
4. View Total Enrollment for Department
5. Quit

Enter choice (1 - 5): 1

----------------------------------------------------

Available Courses
=================

Name:    Introduction to Programming
Course:  CSCI 2000
Section: 60536
Seats:   34

Name:    Intermediate Programming
Course:  CSCI 2003
Section: 63972
Seats:   33

----------------------------------------------------

Options
=======
1. View Available Course Information
2. View Students Enrolled in Course
3. Add Student to a Course
4. View Total Enrollment for Department
5. Quit

Enter choice (1 - 5): 4

----------------------------------------------------

Total Students Enrolled: 3

----------------------------------------------------

Options
=======
1. View Available Course Information
2. View Students Enrolled in Course
3. Add Student to a Course
4. View Total Enrollment for Department
5. Quit

Enter choice (1 - 5): 5
```

```
-----------------------------------------------------

Goodbye!
```

Sample Execution 2: Class is full

```
Course Information Access
=========================

Options
=======
1. View Available Course Information
2. View Students Enrolled in Course
3. Add Student to a Course
4. View Total Enrollment for Department
5. Quit

Enter choice (1 - 5): 1

----------------------------------------------------

Available Courses
=================

Name:    Introduction to Programming
Course:  CSCI 2000
Section: 60536
Seats:   0

Name:    Intermediate Programming
Course:  CSCI 2003
Section: 63972
Seats:   35

----------------------------------------------------

Options
=======
1. View Available Course Information
2. View Students Enrolled in Course
3. Add Student to a Course
4. View Total Enrollment for Department
5. Quit

Enter choice (1 - 5): 3

----------------------------------------------------


Course Options
==============
1. Introduction to Programming
2. Intermediate Programming

Enter choice (1 or 2): 1
```

```
Add a Student
=============

Enter Student's Name: Tyler Greer
Enter Student's ID: 11111111

Class Full!

----------------------------------------------------

Options
=======
1. View Available Course Information
2. View Students Enrolled in Course
3. Add Student to a Course
4. View Total Enrollment for Department
5. Quit

Enter choice (1 - 5): 5

----------------------------------------------------

Goodbye!
```

Requirements:

- Use an updated comment block
- Your program should use the following comment block at the very beginning of your program.

```
// Name:  Your Name                    Date Assigned: Fill in
//
// Course:  CSCI 2003 60357            Date Due:  Fill in
//
// Instructor: Ms. Greer
//
// File name:  Fill in
//
// Program Description:  Brief description of what the program does.
```

- Use appropriate comments throughout the program
- Make good use of whitespace
- Your output should look exactly like the sample output if using the same data.

Deliverables:

- Student.java, Course.java, and CSCIDepartment.java files
- Upload 3 files to Moodle

Total Points	**15 points**
Student class	**2 points**
Name and id instance variables	
Argument and no-argument constructors	
Getters and setters for both instance variables	
Course Class	**9 points**
Name, course code, course section, seats, Student array instance variables	1 point
Enrollment static variable	1 point
Argument and no-argument constructors	0.5 points
Getters and setters for all instance variables	0.5 points
addStudent method	2 points
displayCourseInfo method	2 points
displayStudents method	2 points
CSCIDepartment Class	**4 points**
Creates the two Course objects	1 point
Creates Student objects as needed	1 point
Displays and allows user to choose from the 5 options correctly	0.5 point
Calls methods needed correctly	1 point
Continues until the user enters 5 to quit	0.5 point
Not enough whitespace	**NO DEMO**
Output does not mostly match the sample executions	**NO DEMO**
Bad variable names, method names, and/or class names	**NO DEMO**