# CST 338 – Spring 2017
# Project 1
# Due: 03/10/2017 (Friday) (11:55 PM)

In the project, you will develop four classes called **School**, **Instructor**, **Course**, and **Student** to store instructors, courses, and student information of a school. Based on the sample input data and sample run, you should identify instance variables and methods of each class. If it's necessary, you can add more classes for the project.

## Sample Input Files

The following two sample data files, **C:\\tmp\\test1.txt** and **C:\\tmp\\test2.txt**, will be used in the sample demo programs.

**1. This is a sample file: C:\\tmp\\test1.txt**

```
4
100,Y. Byun,ybyun@csumb.edu,111-111-1111
200,S. Narayanan,sathya@csumb.edu,222-222-2222
300,M. Lara,lara@csumb.edu,333-333-3333
250,S. Bude,bude@csumb.edu,444-123-4567
3
338,CST338 - Software Design,35,BIT 104
205,CST205 - Multimedia Design and Programming,3,BIT 118
306,CST306 - Game Engine Programming,55,BIT 104
2
7777,Alice Otter
8888,Bob Otter
```

The first line (= 4) indicates the number of instructors in the school. The information includes the instructor's unique employee number, name, email address, and phone number. Note that each field of a line is delimited by the comma symbol (,). There's no blank space around the comma symbol and at the end of each line.

After the instructor's data, the number 3 indicates the number of courses in the school. The course data includes the unique course number, course title, course capacity (= max enrollments), and class location.

The last number 2 indicates the number of students in the school. Each student information indicates the student's unique ID and his/her name.

**2. This is another sample file: C:\\tmp\\test2.txt**

```
2
500,G. Bruns,bruns@csumb.edu,555-222-2222
300,O. Doe,doe@csumb.edu,444-333-3333
1
336,CST336 - Internet Programming,3,Library 1010
2
7777,Unknown Otter
9999,John Doe
```

## Some consideration for the project 1:

- For the project, you can assume that the input data file has always correct format and data. For instance, you don't need to check if the email address in the file is valid or not.
- To process the input data with the comma symbol (,) delimiter, **StringTokenizer** class may be useful.
- At the **deleteCourse()** method, a course should not be deleted if there's student(s) enrolled in the course.
- For the **graduateStudent()** method, all courses the student enrolled should be dropped.

### Sample Demo Program 1

The following presents a sample demo program called **SchoolDemo1.java**.

```
public class SchoolDemo1
{
  public static void main(String[] args)
  {
    School SCD = new School("SCD");

    System.out.println("===== Read Data 1 =====");
    SCD.readData("C:\\tmp\\test1.txt");

    System.out.println("\n===== School Info 1 =====");
    SCD.schoolInfo();

    System.out.println("===== Read Data 2 =====");
    SCD.readData("C:\\tmp\\test2.txt");

    System.out.println("\n===== School Info 2 =====");
    SCD.schoolInfo();

    System.out.println("\n===== Search by email =====");
    SCD.searchByEmail("ybyun@csumb.edu");

    System.out.println("\n===== Search by email (fail) =====");
    SCD.searchByEmail("byun@csumb.edu");

    System.out.println("\n===== End of SchoolDemo1 =====");
  }
}
```

A sample run of your program should look like below.

```
===== Read Data 1 =====
Done.

===== School Info 1 =====
School Name: SCD
Instructor Information
       Y. Byun
       S. Narayanan
       M. Lara
       S. Bude
Course Information
       CST338 - Software Design
       CST205 - Multimedia Design and Programming
       CST306 - Game Engine Programming
Student Information
       Alice Otter
       Bob Otter

===== Read Data 2 =====
Instructor info reading failed - Employee number 300 already used.
Student info reading failed - Student ID 7777 already used.
Done.

===== School Info 2 =====
School Name: SCD
Instructor Information
       Y. Byun
       S. Narayanan
```

```
        M. Lara
        S. Bude
        G. Bruns
Course Information
        CST338 - Software Design
        CST205 - Multimedia Design and Programming
        CST306 - Game Engine Programming
        CST336 - Internet Programming
Student Information
        Alice Otter
        Bob Otter
        John Doe

===== Search by email =====
Search key: ybyun@csumb.edu
        Employee Number: 100
        Name: Y. Byun
        Phone: 111-111-1111

===== Search by email (fail) =====
Search key: byun@csumb.edu
        No employee with email byun@csumb.edu

===== End of SchoolDemo1 =====
```

At the sample run, the sequences of instructors, courses, and students are not important. For example, there are three students at the above sample run

```
Student Information
        Alice Otter
        Bob Otter
        John Doe
```

The sequence is not important. If your program displays the sequence in a different order, that's fine.

### Sample Demo Program 2

This is another sample demo program called **SchoolDemo2.java**.

```
public class SchoolDemo2
{
  public static void main(String[] args)
  {
    School SCD = new School("SCD");
    Course course1;

    System.out.println("===== Read Data =====");
    SCD.readData("C:\\tmp\\test1.txt");

    SCD.addInstructor(700, "E. Tao", "tao@csumb.edu", "777-777-1234");
    SCD.addCourse(300, "CST300 - ProSem", 70, "BIT 110");
    SCD.addCourse(499, "CST499 - iOS Dev", 15, "BIT 104");

    SCD.assignInstructor (205, 200);
    SCD.assignInstructor (306, 100);
    SCD.register (306, 7777);
    SCD.register (306, 8888);
    SCD.putScore (306, 7777, 98.54);
    SCD.unRegister (306, 8888);


    System.out.println("\n===== Error Messages =====");
    SCD.addCourse(306, "CST306 - GUI Dev", 25, "BIT 120");
    SCD.putScore (306, 8888, 58.75);
    SCD.register (306, 9999);
    SCD.assignInstructor (499, 900);

    System.out.println("\n===== Detailed Course Info =====");
    SCD.courseInfo(306);

    course1 = SCD.getCourse(205);
    course1.updateLocation("Library 104");
    System.out.println("\n===== Detailed Course Info 2 =====");
    SCD.courseInfo(205);

    System.out.println("\n===== Detailed Course Info 3 =====");
    SCD.courseInfo();

    SCD.deleteCourse(306);
    SCD.deleteCourse(338);
    System.out.println("\n===== Detailed Course Info 4 =====");
    SCD.courseInfo();

    System.out.println("\n===== Good Job! Bye! =====");
  }
}
```

A sample run of your program should look like below.

```
===== Read Data =====
Done.

===== Error Messages =====
Course addition failed – Course number 306 already used.
Student 8888 (Bob Otter) is not enrolled in 306.
Student 9999 does not exist.
Instructor 900 does not exist.

===== Detailed Course Info =====
Course Number: 306
Instructor: Y. Byun
Course Title: CST306 - Game Engine Programming
Room: BIT 104
Total Enrolled: 1
Course Average: 98.54

===== Detailed Course Info 2 =====
Course Number: 205
Instructor: S. Narayanan
Course Title: CST205 - Multimedia Design and Programming
Room: Library 104
Total Enrolled: 0
Course Average: NA   // Or 0 is also fine.

===== Detailed Course Info 3 =====
Number of Courses: 5
       338: 0 enrolled
       205: 0 enrolled
       306: 1 enrolled
       300: 0 enrolled
       499: 0 enrolled

Course deletion failed - Enrolled student(s) in the class
===== Detailed Course Info 4 =====
Number of Courses: 4
       205: 0 enrolled
       306: 1 enrolled
       300: 0 enrolled
       499: 0 enrolled

===== Good Job! Bye! =====
```

### Sample Demo Program 3

This is another sample demo program called **SchoolDemo3.java**.

```
public class SchoolDemo3
{
  public static void main(String[] args)
  {
    School SCD = new School("SCD");
    Instructor instructor1;
    Student student1;

    System.out.println("===== Read Data =====");
    SCD.readData("C:\\tmp\\test1.txt");

    SCD.assignInstructor (306,200);
    SCD.assignInstructor (205,200);
    SCD.addStudent(5555, "Chris Watson");
    SCD.addStudent(9999, "Mike Watson");

    SCD.register (205, 5555);
    SCD.register (205, 7777);
    SCD.register (306, 7777);
    SCD.register (205, 8888);
    SCD.putScore (205, 5555, 50.0);
    SCD.putScore (205, 7777, 100.0);
    SCD.putScore (306, 7777, 100.0);
    SCD.putScore (205, 8888, 50.0);

    System.out.println("\n===== Error Messages =====");
    SCD.register (205, 9999);

    SCD.unRegister (205, 8888);
    System.out.println("\n===== Detailed Course Info 5 =====");
    SCD.courseInfo(205);

    instructor1 = SCD.getInstructor(205);
    System.out.println("\n===== Detailed Instructor Info =====");
    System.out.println(instructor1);

    student1 = SCD.getStudent(7777);
    System.out.println("\n===== Detailed Student Info =====");
    System.out.println(student1);

    SCD.graduateStudent(7777);
    System.out.println("\n===== Detailed Student Info 2 =====");
    System.out.println(SCD.getStudent(7777));

    System.out.println("\n===== Detailed Course Info 6 =====");
    SCD.courseInfo(205);

    System.out.println("\n===== Good Job! Bye! =====");
  }
}
```

A sample run of your program should look like below.

```
===== Read Data 1 =====
Done.
```

```
===== Error Messages =====
Registration failed - Class is full.

===== Detailed Course Info 5 =====
Course Number: 205
Instructor: S. Narayanan
Course Title: CST205 - Multimedia Design and Programming
Room: BIT 118
Total Enrolled: 2
Course Average: 75.00

===== Detailed Instructor Info =====
Instructor Number: 200
Name: S. Narayanan
Courses Teaching:
       306: 1 enrolled
       205: 2 enrolled

===== Detailed Student Info =====
Student Number: 7777
Name: Alice Otter
Courses Enrolled:
       306: 100.00
       205: 100.00
Course Average: 100.00

===== Detailed Student Info 2 =====
Student Number: 7777
No student information.

===== Detailed Course Info 6 =====
Course Number: 205
Instructor: S. Narayanan
Course Title: CST205 - Multimedia Design and Programming
Room: BIT 118
Total Enrolled: 1
Course Average: 50.00

===== Good Job! Bye! =====
```

## <u>Your program will be graded based on</u>

1. Compilation without error.
2. Correct output result.
3. Good programming structure.
4. Comments. (Title, Abstract, Author, ID, and Date are mandatory.)
5. Meaningful and related variable names.

## <u>How to turn in?</u>

Compress all your source files into a single ZIP file. Then, turn it on the **iLearn**. You **should not use** a different compress format such as rar or tar. **We accept only a ZIP file.**