# CS102 - Algorithms and Programming II
# Lab Programming Assignment 6
# Fall 2016

**ATTENTION:**

- Feel free to ask questions on Moodle on the Lab Assignment Forum.
- Compress all of the Java program source files (.java) files into a single zip file.
- The name of the zip file should follow the below convention:
  **CS102_SecX_Asgn6_YourSurname_YourName.zip**
- Replace the variables "YourSurname" and "YourName" with your actual surname and name and X with your Section id (1, 2 or 3).
- Upload the above zip file to Moodle by the deadline before the lab (if not significant points will be taken off). You will get a chance to update and improve your solution by consulting to the TA during the lab. You will resubmit your code once you demo your work to the TA.

**GRADING WARNING:**

- Please read the grading criteria provided on Moodle.

**Q1.** In this lab, you will modify **Classroom** and **ClassroomTester** classes that handles or throws exception. You should carefully inspect the code and think of exceptions that can occur in the program and add code segments accordingly. In these code segments, you will need to use suitable default exceptions classes of Java. If you could not find an appropriate class, you should implement a class, which extends **Exception** class, for that specific exception type.

The files given to you include a **Student** class. A student object has a first and last name, student ID, department ID. Student ID is an integer while department ID is a string of characters (such as CS, EEE,). In addition to the default constructor, it takes student ID, first name, last name and the department respectively in its constructor. Inspect the Student class.

In addition to Student class, there is a class called **Classroom,** which holds an array of students, the name of the class, and the maximum capacity of the class, and the size which is the number of the students who have enrolled. In the constructor, it takes the maximum capacity and the name of the class and initializes the students array with the size of the capacity and sets the number of students to zero. The students in the list are kept in order of their student ID. In Classroom class, the following methods are provided:

- **bestIndex** method that recursively binary searches the students. To do so, it takes a start and end index and also a *student ID*. It checks the middle position, if it is equal to the given ID (matches), it returns the index. Otherwise if the middle is less than the ID, it recursively searches the second half (from middle+1 to end), otherwise it searches the first half (from start to middle -1). If the index does not exist, the start and end will eventually overflow (start > end). This will mark that the index does not exist.
- **addStudent** method that will search for the correct position to insert a student into the list. When you have found the index, shift the students one by one to make room for the new one.

- **removeStudent** method that removes the specified student in the parameter. It looks for the index of the given Student and deletes that Student by simply shifting the Student objects coming after the found index towards to left one position.
- **indexOf** method that returns the index of the given student.
- **searchStudent** method that looks for the student having the given student ID. If it finds, it returns the corresponding Student. Otherwise it returns null. This method makes use of bestIndex method to locate the desired Student.
- **printIntoFile** method that prints out the information of Class properly into the specified file if that file does not exist before.
- **appendToFile** method that appends the information of Class properly to the specified file if that file exists.
- **toString** method that converts the information of Class into a String and returns this variable.

To demonstrate the exceptions you handle, you should fill the main method of **ClassroomTester** class. In this method, create Class object(s) and Student objects, and make use of these objects to test the exceptions.