CS2013 Course Syllabus  
Spring 2018  
Section 7/8: Lecture Mon/Wed 9:00 A.M. – 9:50 A.M. KH B2007  
Lab: Mon/Wed 9:50 A.M. – 11:00 A.M. KH B2007  

Instructor  
Jungsoo (Sue) Lim,  
Email: jlim34@calstatela.edu  
Office hours:  
1. Mon/Wed: 11:30 AM – 12:30 PM at E&T A310  
2. Tue/Thu: 12:00 Noon – 1:00 PM at E&T A310  
3. Tue/Thu: 4:30 PM – 5:30 PM at E&T A310  
4. By Appointment  

Course name  
Programming with Data Structures  

Credits  
3 units  

Contact hours  
10 hours/week (2 hours lecture, 2 hours lab, 6 hours workshop)  

Text book  
Required: Introduction to JAVA Programming, by: Daniel Liang, 10th Edition  

Course Information  
a) This course will provide an accelerated version of CSULA’s three-quarter introductory Java programming sequence.  
b) The major topics of the course include an introduction to programming using Java, introduction to Object-Oriented Programming, designing and coding programs.  
c) Laboratory activities on problem analysis and software development.  
d) Prerequisites: Math 103  
e) Recommended Prerequisite: Math 206: Calculus I  

Course Goals  
The Student Learning Outcomes that are addressed by the course are:  
SLO #2: Students will be able to demonstrate fluency in at least one programming language and acquaintance with at least three more.  
SLO #3: Students will have a strong foundation in the design, analysis, and application of many types of algorithms.  
SLO #5. Students will have the training to analyze problems and identify and define the computing requirements appropriate to their solutions.  
SLO #6. Students will have the training to design, implement, and evaluate large software systems working both individually and collaboratively.  
SLO #7. Students will be able to communicate effectively orally and in writing.
Other Outcomes of Instruction
At the end of the course students are able to:

- Divide a problem into its logical set of components
- Have a good understanding of the object-oriented programming concepts
- Create multiple classes to represent objects in the program definition.
- Have a good understanding of inheritance and polymorphism.
- Design and code high-level GUI programs.
- Use recursion as a tool to solve some specific problems.
- Know the standard Abstract Data Types, and their implementations.
- Study and use different available JAVA Data Structures.
- Know the standard searching and sorting algorithms and their efficiency.
- Understand the complexity analysis for some simple software.

Brief list of topics to be covered

- Mathematical functions, Characters, and Strings
- Selections, Loops, and Methods
- Single-Dimensional and Multi-Dimensional Arrays
- Objects and Classes
- Defining Classes for Objects
- Constructing Objects Using Constructors
- Using classes from Java Library
- Visibility Modifiers
- Passing Objects to Methods
- Array of Objects
- Thinking in Objects
- Inheritance and Polymorphism
- Exception Handling
- Event-Driven Programming
- Creating Graphical User Interfaces
- Recursion
- Lists, Stacks, Queues, and Priority Queues
- Binary Search Trees

Laboratory Projects
Each week students will complete a 2-hour lab assignment and 6-hour workshops on selected topics. At the end of each lab and workshops students will turn in programming assignments.

Out of class Assignments
Each week students will have an assignment due on the following week, except the exam weeks. For these assignments students may be required to complete an unfinished implementation, design/implement a system, produce system document (pseudo code, UML, etc), and prepare a user’s manual.

Quizzes
Each workshop class will have in-class quiz.
**Grading Policy for lecture part**

Class Attendance: 10%

Lab Assignments: 20%

4 Midterms: 40% (4x10%)

Final: 30%

<table>
<thead>
<tr>
<th>Grade</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>A-, A</td>
<td>90 – 100</td>
</tr>
<tr>
<td>B-, B, B+</td>
<td>80 – 89</td>
</tr>
<tr>
<td>C-, C, C+</td>
<td>65 – 79</td>
</tr>
<tr>
<td>D, F</td>
<td>&lt;65</td>
</tr>
</tbody>
</table>

**Academic Integrity**

Cheating will not be tolerated. Cheating on any assignment or exam will be taken seriously. All parties involved will receive a grade of F for the course and are reported to the proper authorities.

**ADA Statement**

Reasonable accommodation will be provided to any student who is registered with the Office of Students with Disabilities and requests needed accommodation.
## CS2013 Course Weekly Schedule

<table>
<thead>
<tr>
<th>Week</th>
<th>Lecture topics</th>
</tr>
</thead>
</table>
| 1    | - Introduction to Computers, Programs, and Java, Elementary Programming (Ch 1 – 2)  
     | - Selection – Boolean data type, logical operators, switch statement, and conditional expression (Ch 3)  
     | - Overview of Eclipse IDE  
     | 2    | - Mathematical Functions, Character, and Strings (Ch 4)  
     |      | - Loops – while loop, do-while loop for loop, break and continue (Ch 5)  
     | 3    | - Method – defining, calling, passing arguments, and overloading method (Ch 6)  
     |      | - Arrays – Single-Dimensional array and multi-dimensional array (Ch 7 – 8)  
     |      | **Midterm 1 (Ch 1 – 5)**  
     | 4    | Objects and Classes (Ch 9)  
     |      | - Defining Classes for Objects  
     |      | - Constructing Objects Using Constructors  
     |      | - Using classes from Java Library  
     |      | - Visibility Modifiers  
     |      | - Passing Objects to Methods  
     | 5    | Thinking in Objects (Ch 10)  
     |      | - Class Abstraction and Encapsulation  
     |      | - Class Relationships  
     |      | - Processing Primitive Data Type Values as Objects  
     | 6    | Inheritance and Polymorphism (Ch 11)  
     |      | - Super classes and Subclasses  
     |      | - Overriding Methods  
     |      | - Polymorphism  
     |      | - Dynamic Binding  
     |      | **Midterm 2 (Ch 6 – 10)**  
     | 7    | Exception Handling and Text I/O (Ch 12)  
     |      | - Exception Handling Overview  
     |      | - Exception Types  
     |      | - File Input and Output  

Page 4 of 6
<table>
<thead>
<tr>
<th>8</th>
<th>Abstract Classes and Interfaces (Ch13)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• Abstract Classes</td>
</tr>
<tr>
<td></td>
<td>• Interfaces</td>
</tr>
<tr>
<td>9</td>
<td>Graphics - GUI (Ch 14)</td>
</tr>
<tr>
<td></td>
<td>• Basic Structure of JavaFX</td>
</tr>
<tr>
<td></td>
<td>• Panes, UI Controls, and Shapes</td>
</tr>
<tr>
<td></td>
<td>• Property Binding</td>
</tr>
<tr>
<td></td>
<td><strong>Midterm 3 (Ch 11 – 13)</strong></td>
</tr>
<tr>
<td>10</td>
<td>Event-Driven Programming (Ch 15)</td>
</tr>
<tr>
<td></td>
<td>• Event and Event Sources</td>
</tr>
<tr>
<td></td>
<td>• Registering Handlers and Handling Events</td>
</tr>
<tr>
<td></td>
<td>• Anonymous inner Class Handlers</td>
</tr>
<tr>
<td>11</td>
<td>Creating Graphical User Interfaces – Label, Button, CheckBox, RadioButton, etc (Ch 16)</td>
</tr>
<tr>
<td></td>
<td>Binary I/O – binary I/O, Object I/O, Random-access files (Ch17)</td>
</tr>
<tr>
<td>12</td>
<td>Recursion (Ch 18)</td>
</tr>
<tr>
<td></td>
<td>• Describe what a recursive method is and the benefits of using recursion</td>
</tr>
<tr>
<td></td>
<td>• Develop recursive method</td>
</tr>
<tr>
<td></td>
<td>• Review how recursive method calls are handled in a call stack</td>
</tr>
<tr>
<td></td>
<td><strong>Midterm 4 (Ch 14 – 17)</strong></td>
</tr>
<tr>
<td>13</td>
<td>Recursion (Ch 18) – continued</td>
</tr>
<tr>
<td></td>
<td>• Solve problems using recursion</td>
</tr>
<tr>
<td></td>
<td>• Implement a selection sort using recursion</td>
</tr>
<tr>
<td></td>
<td>• Implement a binary search using recursion</td>
</tr>
<tr>
<td>14</td>
<td>Generics (Ch 19)</td>
</tr>
<tr>
<td></td>
<td>• Define generic classes and interfaces</td>
</tr>
<tr>
<td></td>
<td>• Define and use generic methods and bounded generic types</td>
</tr>
<tr>
<td></td>
<td>• Develop a generic sort method to sort an array of Comparable objects</td>
</tr>
<tr>
<td>15</td>
<td>Implementing Lists, Stacks, Queues, and Priority Queues (Ch 24)</td>
</tr>
<tr>
<td></td>
<td>• Design common features of lists in an interface and provide skeleton implementation in an abstract class</td>
</tr>
<tr>
<td></td>
<td>• Design and implement an array list using an array</td>
</tr>
<tr>
<td></td>
<td>• Design and implement a linked list using a linked structure</td>
</tr>
</tbody>
</table>
- Design and implement a stack class using an array list and queue class using a linked list
- Design and implement a priority queue using a heap

**Binary Search Trees (Ch 25)**
- Design and implement a binary search tree
- Search an element in a binary search tree
- Insert/delete an element into a binary search tree
- Traverse/display elements in a binary tree (in-order, pre-order, post-order, breath first traversal)

| 16 | **Final Exam** |