OCS25: Data Structures and Algorithms in Java

Course Description
Data Structures and Algorithms continues and deepens students' understanding and practice of object oriented programming. The course is intended for students who have familiarity with programming in Java at AP Computer Science level A and give the prerequisite computation skills needed in computer science or engineering.

Core topics in the context of the Java programming language include practical implementations of fundamental and more advanced data structures (linked lists, hash encoded storage, binary search trees - AVL, treaps, red-black trees, and heaps), algorithms for organizing and manipulating data (including sorting, searching, and traversal algorithms), and time complexity of algorithms in a problem-solving oriented context. In-depth exploration of standard Java libraries and features such as Java Collections, error handling, threads, and designing and building graphical user interface using AWT and Swing libraries is included. Much of the course is project-based, with assignments stressing the design of classes and algorithms appropriate to a particular problem.

Learning Objectives
Upon completion of this course students will be able to:

- Use object oriented techniques to solve complex problems.
- Use object-oriented program design including abstraction, encapsulation, information hiding, and object relationship including composition, containment, inheritance, cohesion and coupling.
- Learn the concepts abstract data structures, analyze the complexity of algorithms, and design and implement efficient code.
- Understand the most commonly used algorithms for sorting and searching, including balanced search trees, hash tables, minimum spanning trees, and shortest paths.
- Understand the basic string operations including sorting, searching, regular expression, and string compression.

Required Textbook
Algorithms
Robert Sedgewich and Kevin Wayne

Course Topics
- **Introduction**
  Data abstraction, linked data structures, implementation of stacks, queues, bags.
- **Analysis of Algorithms**
  Time complexity and memory usage of algorithm and order of growth of functions
- **Sorting**
  Elementary sorting techniques, merge sort, quicksort and priority queues
- **Searching**
  Symbol tables, binary search trees, 2-3 search trees, Red-black binary search trees
• **Graphs**
  Undirected and directed graphs, minimum spanning trees including Prim’s and Kruskals algorithms, and shortest paths algorithm including Dijkstra’s algorithms

• **Strings**
  String sorting techniques, symbol tables, substring searches, regular expression, and data compression

**Overview of Assignments**

• **In class participation:** You are expected to participate in in-class discussions and discussions on the course webpage and be part of an active learning environment. This portion of your grade will be determined based on your attendance, regular participation in the discussion including asking and answering questions, and work (usually in small groups) on in-class labs.

• **Quizzes, homework and programming labs:** The primary homework will be programming assignments. In addition to programming assignments, there will be occasional worksheets or other written assignments.

• **Midterm:** There will be a two-hour proctored midterm exam.

• **Final Exam:** There will be a three-hour proctored final exam.