COMPUTER SCIENCE B: DATA STRUCTURES

SYLLABUS 2023 - 2024

COURSE DESCRIPTION

Prerequisites: AP Computer Science A or a 3 or higher on the AP CSA Exam

Computer Science B - Data Structures is a college-level course that covers the design, development, testing, and debugging of computer programs using abstract data types in the object-oriented programming language Java. The course is designed to serve students who have prior computing experience equivalent to AP Computer Science A. Data structures and algorithms are concepts that are language independent, and emphasis will be placed on the study of the following abstract data types: lists, stacks, queues, linked structures, binary search trees, hashing, graphs, searching and sorting using recursive algorithms

REQUIRED TEXTBOOK

- Goodrich, M., Tamassia, R., & Mount, D. (2011). Data Structures and Algorithms. Hoboken, NJ: Wiley

GRADING SCALE

Your final grade will be weighted according to the following categories. Each assessment, quiz, or project will be graded according to a specific rubric.

Class Assignments	**20%**	Coding Responses, Written Responses, Small Programs
Assessments	**40%**	Program Code, Written Responses, Code Segments, Multiple Choice, Code Evaluation
Quizzes	**20%**	Multiple Choice, Short Code, Written Responses
Projects	**20%**	Development Process & Logs, Written Components, Larger Programs with Applications

GRADING SCALE

A+ 97% - 100%	**B+ 87% - 89.99%**	**C+ 77% - 79.99%**	**D+ 67% - 69.99%**	**F 0% - 65%**
A 93% - 96.99%	**B 83% - 86.99%**	**C 73% - 76.99%**	**D 63% - 66.99%**	
A- 90% - 92.99%	**B- 80% - 82.99%**	**C- 70% - 72.99%**	**D- 60% - 62.99%**	

CLASS ASSIGNMENTS & HOMEWORK

Assignments may include reading, written responses, or short coding programs or segments. In class exercises are to be completed in class and are graded as complete/incomplete and are due by the start of the next class period. Any unfinished class assignment will become homework. Incorrect assignment submissions such as missing portions, incomplete sentences, will not be accepted however revisions and late work can be accepted due to absences or on a case by case basis.

ACADEMIC HONESTY

Irvine Unified School District places an emphasis on and promotes ethical behavior and academic honesty. An environment of academic honesty ensures student learning is assessed accurately, and honest hard-working students are not at a disadvantage. Academic dishonesty will not be tolerated in any form. Your responsibilities are outlined in the pamphlet, IUSD Commitment to Academic Honesty. The following activities will also be considered dishonest:

- Turning in anyone else's work as if you did the work yourself.
- Sharing any materials during a test.
- Using online resources without citing sources (when allowed)
- Allowing another student to look at your work or looking at another student's work.

Because you often learn class in collaborative study teams, confusion sometimes arises regarding copying. In general, you should be doing all assignments on your own. Copying another student's work, or allowing another to copy your work, is cheating and is dishonest. Copying involves seeing another student's work and replicating it. On the other hand, Collaboration is sharing information, expertise, and opinions. You are allowed to discuss approaches and processes freely, but you are not allowed to show your work. As a general rule of thumb, if you have a question, you should NEVER look at another student's solution, instead, you can have a student who has already completed the solution look at your work and troubleshoot.

COURSE OUTLINE

<u>Unit 1</u>: Java Review	· Selection Statements · Loops · Classes ·Constructors ·Variables/Instance Variables · Methods · Inheritance · Abstract Classes · Nested Classes · Exception Handling · File Writing · GUI/Android Studio
<u>Unit 2:</u> Fundamental Data Structures	· One dimensional arrays · Two dimensional arrays · Singly linked lists · Circularly linked lists · Doubly linked lists · References vs. values

	· Cloning · Equivalence testing · JSON/Google Firebase
Unit 3: Recursion and Algorithm Analysis	· Recursion: linear, binary, and multiple · Tracing and writing recursive methods · Comparing growth rates · "Big Oh" Notation · Analyzing algorithms
Unit 4: Sequential Data Structures	· Stacks · Queues · Double Ended Queues (Dequeues) · List ADT · Dynamic Arrays · Iterators
Unit 5: Trees	· General Trees · Depth and height · Tree ADT · Binary Trees · Traversing Trees
Unit 6: Priority Queue ADT	· Heaps · Sorting with a Priority Queue · Adaptable Priority Queue
Unit 7: Map, Hash Tables, and Skip Lists	· Maps · Hash Tables · Sorted Maps · Skip Lists · Sets · Multimaps · Multisets
Unit 8: Search Trees	· Binary Search Trees · Balanced Search Trees · AVL Trees · Splay Trees · (2,4) Trees · Red-Black Trees
Unit 9: Graph Algorithms	· Graphs · Graph Traversals · Directed Graphs · Weighted Graphs · Shortest Path Algorithms · Minimum Spanning Trees · Prim-Jarnik Algorithm · Kruskal's Algorithm