四川大學
SICHUAN UNIVERSITY

Programming Data Structures

Course Number:	CS 225	**Term:**	Summer, 2021
Instructor:	TBA	**Email:**	
Contact Hours:	48	**Meeting Times:**	TBA
Credits:	3.0		

Course Description:

This introductory course in data structures will provide a solid foundation of practical and theoretical knowledge. The course will include coverage of the definition, design, and implementation of data structures, including arrays, stacks, queues, heaps, and linked structures. Coverage of structures will include various types of hash tables, trees, and graphs. Additionally, the course will thoroughly examine algorithms for manipulating these structures for searching and sorting, including introducing graph algorithms and analyzing sorting and searching algorithms.

Learning Objectives:

Upon successful completion of this course, students will be prepared to:

1. Store and access data with fast and efficient algorithms
2. Evaluate efficiency and speed of code
3. Determine appropriate data structures
4. Apply common data structures and algorithms
5. Implement data structures in various programming languages

Required Textbook and Course Materials:

Data Structures and Algorithms
Alfred V. Aho, Jeffrey D. Ullman and John E.
Hopcroft, Addison-Wesley
ISBN 0-201-00023-7

Language of Instruction:

This course is taught entirely in English, including lectures, homework/other assignments and examinations. Teaching assistants will be fluent in both English and Mandarin.

Course Prerequisites

CS 201 / CS 202 / CS 203 Introduction to Programming or equivalent

---

University Policies

Class Format

In Person. Course activities, discussions, assignments and resources will be made available at the start of and during the course.

Attendance, Participation and Deliverables

Courses are very intensive and in order to be successful, students need to attend every class. Attendance is required for all lectures and class activities. Class participation is expected from every student and form a significant portion of the final course grade.

All course deliverables (homework assignments and tests) are due on time as assigned. This course includes *no* make-ups, postponements or additional assignments, except for verified medical emergencies. If you miss an exam/assignment due to a non-sanctioned absence, your score on that exam/assignment will be zero

Academic Dishonesty

All cases of academic dishonesty will be diligently pursued. Academic dishonesty includes representing the work of another as one’s own work or cheating by any means. Academic dishonesty also includes aiding, abetting, concealing or attempting such activity. The penalty is automatic failure of the course and possible suspension from the university.

Grading Scale (%)

97-100	A+
93-96	A
90-92	A-
87-89	B+
83-86	B
80-82	B-

77-79	C+
73-76	C
70-72	C-
67-69	D+
63-66	D
60-62	D-
0-59	F

Professor- and Course-Specific Policies (*Tentative*)

Homework

Assignments will be listed at the beginning of the course. The purpose is to prepare you for the exams. The homework is a very important part of the course. No matter how well you think you understand the material presented in class, you won't really learn it until you do the problems.

Exams

No make-ups will be given after the exam. The use of the textbook or any other written reference is not allowed during the exams. The purpose of the exams is to test your understanding of key concepts from the course lectures and materials.

Grade Components

Quizzes	20%
Assignments	20%
Mid Term Exam	25%
Final Exam	35%
Total	**100%**

Course Schedule (*Tentative*)

Module	Topics
1	**Data and Algorithmic Concepts in Software** • Define Fundamentals of Software & Software Practice • Data Types • Abstract Data Types • Algorithmic Fundamentals **Software Metrics** • Rationale for Metrics • Big Oh Notation • Graphic Representations • Other kinds of metrics
2	**Array & Indirection** • Pointers and Indirection for Data Structures • Introduce UML notation for Data Structures • What is an Array • Array traversals (e.g.., Binary search, etc.) **Stacks** • Stack specifications • Stack behaviors • UML Representation of Stack • Stack implementations • Examples of Stacks in Software Design
3	**Queues** • Queue specifications • Queue behaviors (more than one kind of queue) • UML Representation of Queue • Queue implementations **Heaps & Matrices** • Heap algorithms • Matrix Specifications • Matrix behaviors (Two dimensional) • UML Representation of Matrix • Matrix implementations • Examples of Matrix in Software Design

4	**Sorting Algorithms** • Bubble • Insertion • Quicksort • Other kinds of sorting with associated metrics **Trees** • Binary • Tree algorithms • AVL versus Red-Black • Depth Search • Breadth Search
5	**Hash Tables** • Continue with Trees if necessary • Hash tables **Graphs** • Overview of Graphs and Graph Theory • Undirected Graphs • Undirected Graph Algorithms • Begin on Directed Graphs • Directed Graphs • Directed Graph algorithms Final Exam