



CSC 022: Data Structures

Term: Summer, 2019

Time: Monday – Friday, 5 Weeks

Class Sessions per Week: 5

Class Session Length (minutes): 150

Total Contact Hours: 60

Course Description

This course teaches different techniques of storing, accessing and processing data as utilized in the development of programs and algorithms. Topics include linked lists, stacks, queues, trees, recursion and graphs. Algorithms for applications such as sorting, searching and merging will be analyzed and implemented. Solutions are designed using object-oriented techniques and implemented in the C++ programming language. This course is the third course in the C++ programming sequence and is a required course for students in the Computer Information Systems General Option.

Learning Outcomes

1. Demonstrate the ability to choose the appropriate data structures when solving typical computer science problems.
2. Demonstrate the ability to sort, search, and merge data using an object oriented programming language.
3. Demonstrate the ability to effectively develop and implement customized data structures.
4. Analyze algorithms to determine time and space complexity.

Textbook(s)

The C++ Programming Language, 4th Edition, Bjarne Stroustrup (May 2013)

Course Resources



Code Viewers/Editors:

- Sublime Text
- Atom
- Emacs

C++ IDEs:

- Eclipse
- wxDev-C++
- XCode (MaxOSX only)

Compilers:

For this course, any C++ compiler is accepted. However, the submitted homework code must compile and run on gcc/g++ 4.8.X.

Note on C++ versions:

Although all operating systems support the new features in C++11, for this course, we are avoiding the new features in order to have better practice.

Course Requirements

Homework and lab: For every week, there will be homework and lab. The recitation is mandatory and will take attendance. There will be three exams in this semester. The final exam will be cumulative, covering all the material covered in class.

Grading

Homework	10%
Lab assignments	20%
Unit Exam (20% per exam)	40%
Final Exam	30%
Total	100%

A: 90-100 A-: 85-89 B+: 82-84 B: 78-81 B-: 75-77 C+: 72-74 C: 68-71 C-: 64-67 D: 60-63 F: <60

Schedule

Week 1

Introductions and basic facilities

Recursion

Week2



Analysis of Algorithms

- Order of complexity
- Big O
- Worst case and average case analyses

List ADT and Applications

- implementation with dynamic storage
- implementation with static storage
- iterative and recursive list operations
- singly, doubly and circularly linked lists

Exam 1

Week3

Stack ADT and Applications

- push, pop, full, empty operations
- implementation using list

Queue ADT and Applications

- add, delete, empty, full operations
- implementation using list

Week4

Sorting and Searching linear search

- binary search
- selection sort
- insertion sort etc.

Binary Tree ADT and Applications representation

- tree traversals (recursive and non-recursive) binary search trees

Exam 2

Week 5

More about Binary Trees

- AVL trees
- B- trees

Graphs and Applications

- definition and representation of graphs and digraphs graph application
(reachability, connected components, shortest path)

Final Exam

Policies



Exam and Attendance

No make-ups for missed exams. If legitimate absence, midterm exam can be excused. Attending all class meetings is required. Students are responsible for missing class by any reason. We will be covering a lot of material quickly so it is detrimental to miss even a day.

Academic Honesty

Xihua University expects all students to do their own work. Instructors will fail assignments that show evidence of plagiarism or other forms of cheating, and will also report the student's name to the University administration. A student reported to the University for cheating is placed on disciplinary probation; a student reported twice is suspended or expelled.

Special needs

Students with disabilities and special needs should consult with the professor early in the semester. Recording of lectures is permitted.