

CS 1521: Computer Science II (5)**Catalog Description:**

Continuation of introduction to computer science. Methods for procedural and data abstraction. Focus on classical data structures, procedural and data abstraction, and the abstract data type. Introduction to software engineering technique. Algorithm analysis, principles of object-oriented programming issues in ethical use of computers. Requires implementation of significant programming projects.

Textbook: Caranno, Frank M. *Data Abstraction and Problem Solving with C++: Walls and Mirrors*. 5th ed. Addison Wesley, 2007.

Course Goals:

The course begins with an introduction to object-oriented design (OOD) including: the Unified Process (UP) development cycle, iterative development techniques, use case analysis, Unified Modeling Language (UML) sequence and static diagrams. The basic principles of object-oriented programming (OOP) encapsulation, information hiding, inheritance, and polymorphism, are taught. Discussion of the importance of procedural and data abstraction leads to the concept of an ADT (Abstract Data Type). The ADT concept is then elaborated on in a series of assignments and lectures covering the basic ADTs: lists, stacks, queues, trees, priority queues, tables, and graphs. Advanced C++ programming techniques are taught and practiced. Student understanding of recursion and algorithm analysis is extended. By the end of the course, a student should have mastered the main concepts of OOD, OOP, and have successfully completed major programming assignments in C++ on each of the basic ADTs.

Prerequisites by Course & Topic

CS 1511: Computer Science I or CS 1581: Honors: Computer Science I – Working knowledge of the C++ language, basics of software design, basics of algorithm analysis, introduction to computer ethics

Major Topics Covered in the Course

- Advanced features of the C++ language
- Object-oriented design principles
- Algorithmic analysis tools, focusing on Big-O, including analysis of recursive algorithms
- Introduction to Abstract Data Types, including: lists, stacks, queues, trees, tables, priority queues, and graphs
- Intermediate searching and sorting routines, including: binary search, mergesort, quicksort, radix sort, tree sort, and heapsort

Class/Laboratory Schedule: Lecture: 3 hrs per week, Discussion: 1 hr per week, Laboratory: 1 hr per week

Course Outcomes

1. Students are introduced to object-oriented analysis and design and have a firm foundation in object-oriented programming in the C++ language.
Students have the ability to read and understand use cases, Unified Modeling Language (UML) static diagrams. Given a design, students can implement that design or a portion of it in C++. In particular, they are able to develop and test efficient abstract data types (ADT) as objects, and are able to use those ADTs to solve larger problems.
2. Students have the ability to analyze the run-time characteristics of any given algorithm.
Students have the ability to analyze the asymptotic efficiency (Big-Oh) of the operations associated with an ADT, and search and sort algorithms. This includes analysis of both iterative and recursive routines.
3. Students understand ethical issues in software.
Understand the implications that various asymptotic efficiency characteristics of an operation will have on client code.

Relationship to Program Outcomes

CS 1521 is the second course in the introductory computer science sequence. This course contributes to meeting the following program outcomes:

2. Students can design, develop, and analyze significant software systems.

Course outcome 1 gives CS 1521 students an introduction to use case analysis and basic UML design diagrams. Solution designs requiring ADTs are given to students to implement.

Course outcome 2 gives students experience analyzing the asymptotic efficiency of the algorithms utilized in ADT operations. Both iterative and recursive versions of many solutions are developed and analyzed.

3. Students understand the fundamentals of computer organization and architecture, data structures and related algorithms, and programming languages.

Course outcome 1 gives CS 1521 students a focus on ADTs: list, stack, queue, tree, table, heap, priority queue, and graph and their related algorithms. As well as advanced knowledge of the C++ language to include: implementing UML design diagrams (generalization, containment, implemented-in-terms-of), techniques for dynamic memory management (new and delete operators, constructors, destructors, overloaded assignment operator), file I/O, STL string class, inheritance (public and private), static and dynamic binding (virtual methods), abstract base classes (pure virtual methods), friend functions and friend classes, class templates, operator overloading.

Course outcome 2 gives students experience with search and sort algorithms, including: binary search, selection sort, insertion sort, bubble sort, merge sort, quick sort, radix sort, tree sort, and heap sort.

7. Students understand social, professional and ethical issues related to computing.

Course outcome 3 gives CS 1521 students an understanding of the importance of data organization and asymptotic algorithm efficiency as it relates to time-critical operations (worst-case versus average-case analysis).

Assessment Plan for Course:

This course is assessed every other year by the instructor and a course assessment document covering all of the course outcomes and their effect on the program outcomes is prepared.

Estimate CSAB Category Content

	CORE	ADVANCED		CORE	ADVANCED
Data Structures	2		Computer Organization and Architecture	0	
Algorithms	0.5		Concept of Programming Languages	1.5	
Software Design	1				

Coordinator/Prepared by: S. Holtz