

## SYLLABUS

### *CS 162: Introduction to Computer Science*

Credit Hours:	4
Course Coordinator:	Karla Fant
Course Description:	Introduction to programming using a high level programming language. Conditionals, I/O, Files, Functions, Classes, Pointers, Dynamic Memory, Linear Linked lists, and Multi-Dimensional Arrays. Program correctness, verification, and testing. Prerequisite: CS106, or CS161, or prior programming experience.
Prerequisites:	CS 106, or CS 161, or prior programming experience.
Goals:	The goals of this class are to teach the syntax of a high level programming language to students who already know what programs are. This course will introduce the syntax of C++ including: data types, variables, conditionals, loops, functions, and arrays. It will introduce new constructs such as classes, pointers, dynamic memory, linear linked lists, recursion, and multi-dimensional arrays to prepare students for CS163, Data Structures. To learn and use advanced concepts provided by a modern structured computer language, such as data abstraction, separate compilation and the use of library procedures.
Textbooks:	Russell Shackelford, <i>An Introduction to Computing &amp; Algorithms</i> , Addison-Wesley.
References:	Optional: Weekly Lecture Notes and Course Slides
Major Topics:	Introduction and overview (4 hours) I/O, Conditionals, Repetition, Arrays (4 hours) Functions and Arguments (4 hours) Introduction to Classes, Constructors, Separate Files (4 hours) Introduction to Pointers and Dynamic Memory (8 hours) Introduction to Linear Linked Lists (4 hours) Recursion (4 hours) File I/O (4 hours) Arrays with Structured Elements (4 hours)
Laboratory Exercises:	5 programming assignments to be turned in via email on the specified due date. 4 written homework assignments. Each programming assignment includes a written algorithm. 3 of the homework's include a discussion of social and/or ethical implications.

CAC Category Credits	Core	Advanced
Data Structures	1	
Algorithms	1	
Software Design		
Computer Architecture		
Programming Languages	2	

Oral and Written Communications:	Every student is required to submit at least five written algorithms with each programming assignment which are typically between one and two pages in length. This material is graded for grammar, spelling, style, technical content, appeal, completeness, and accuracy.
Social and Ethical Issues:	Students will study a collection of ethical case studies involving computers and investigate how these apply to databases of private information. Students will analyze the security of their programs as part of the homework and discussions.
Theoretical Content:	Students are introduced to the C++ programming language along with program design, testing, and debugging. Emphasis is placed on the notion of programming paradigms: moving from procedural, modular, to data abstraction. Dynamic memory management is introduced along with linear linked lists, recursive function design and implementation.
Problem Analysis:	Discussions focus on analysis and design issues on how to make programs complete, and user friendly while incorporating concepts such as files, classes, linked lists, searching, and recursive methods.
Solution Design:	Students gain experience by writing algorithms and designing software to solve the programming projects assigned in class.

---