Intro to CS Syllabus

Fall 2017

# Instructor Information

|  |  |  |
| --- | --- | --- |
| Instructor | Email | Office Location & Hours |
| **Denny Anderson** | [dra2zp@virginia.edu](mailto:dra2zp@virginia.edu) | TBA |

# General Information

## Description

This course serves as an introduction to computer science. Many topics will be taught in this class, including basic mathematics, the fundamentals of programming (relying heavily on the C++ computer language with either the clang++ or the gcc compiler), statements and loops, computer architecture, parameter passing, software development, computer security, and algorithms.

## Expectations and Goals

All students will be expected to follow the University of Virginia’s Honor Code at all times. Collaboration with other students is prohibited on homework assignments and in-class quizzes. Using notes is prohibited on in-class quizzes. Reviewing notes is strongly encouraged on homework assignments. If a student is having difficulty, they will be expected to contact the instructor. The Honor Code will be enforced on every assignment. If a student violates the Honor Code, they will be dismissed from the course with a grade of “F.”

## Grading Scale

97.0% A+ 93.0% A 90.0% A-

87.0% B+ 83.0% B 80.0% B-

77.0% C+ 73.0% C 70.0% C-

67.0% D+ 63.0% D 60.0% D-

0.0% F

A grade greater than or equal to 60.0% is considered passing. A grade less than 60.0% is considered failing.

## Grade Weights

Attendance 10%

Homework 50%

Quiz 10%

Final Exam 30%

# Course Materials

## Required Materials

The only materials that this class requires is a laptop with Internet access. The website Code Chef (<https://www.codechef.com/ide>) will be used frequently for coding assignments, and students will be expected to create an account before the second class meeting.

## Required Text

No textbook is required for this course.

# Course Schedule

**Chapter 1: Introduction to Programming** **Chapter 5: Parameter Passing**

Lesson 1: Computer Science Lesson 1: Passing by Value

Lesson 2: Computer Programming Lesson 2: Passing by Reference

Lesson 3: Programming Languages Lesson 3: Pointers

Lesson 4: Introduction to C++ Lesson 4: Null Pointers

Lesson 5: Reserved Words Lesson 5: Dereferencing a Pointer

Lesson 6: Return Types Lesson 6: Dangling Pointers

Lesson 7: Hello, World! Program Lesson 7: Segmentation Faults

**Chapter 2: Mathematics and Programming** **Chapter 6: Software Development**

Lesson 1: Functions Lesson 1: Requirements

Lesson 2: Input and Output Lesson 2: Design

Lesson 3: Parameters Lesson 3: Coding/Implementation

Lesson 4: Function Calling Lesson 4: Integration

**Chapter 3: Statements and Loops** Lesson 5: Release

Lesson 1: if Statements Lesson 6: Maintenance

Lesson 2: if / elif / else **Chapter 7: Computer Security**

Lesson 3: Iteration Lesson 1: Vulnerabilities

Lesson 4: Arrays and Vectors Lesson 2: (Stack) Buffer Overflow Attacks

Lesson 5: Elements and Indexing Lesson 3: Address Space Layout Randomization

Lesson 6: for Loops Lesson 4: Stack Canaries

Lesson 7: while Loops Lesson 5: Pointers

**Chapter 4: Computer Architecture** Lesson 6: Viruses

Lesson 1: Compiling Lesson 7: Sandboxing with VirtualBox

Lesson 2: Linking Lesson 8: Vector [index] vs. .at(index)

Lesson 3: Assembly **Chapter 8: Algorithms**

Lesson 4: Machine Code Lesson 1: Properties of an Algorithms

Lesson 5: Memory Layout Lesson 2: Examples of Algorithms

Lesson 6: The Heap Lesson 3: Adding Array Elements

Lesson 7: The Stack Lesson 4: Sorting

Lesson 8: Cache Lesson 5: Searching