

# COMPUTER SCIENCE (CS)

## CS-115 Introduction to Problem Solving and Programming

### 3 Credits

**Lecture:** 3 hours per week

**Offering:** Fall, Spring, and Summer, All Years

This course provides an introduction to computational thinking and problem solving. Students will be able to apply elementary computing concepts including variables, loops, functions, lists, conditionals, concurrency, data types, simple object oriented concepts, I/O, events, syntax, and structured programming. Basic concepts of computer organization and editing, and the influence of computers in modern society will be explored. NOTE: CS-115 carries no credit if taken after successful completion of higher numbered computer science courses.

**Prerequisites:** MATH-108 or an appropriate score on a placement test.

## CS-150 Computer Science I

### 4 Credits

**Lecture:** 3 hours per week, **Lab:** 2 hours per week

**Offering:** Fall and Spring Only, All Years

This course offers an introduction to the field of computer science using a current programming language. Central themes of the class include an introduction to computer organization; algorithmic problem solving; structured and object oriented program design; and the societal and professional context in which computer science exists. Fundamental data types including arrays and structures will be explored and concepts such as sorting and searching algorithms, complexity, invariants, abstract data types and pointers will be introduced.

**Corequisites:** CS-150L

**Pre/Corequisites:** MATH-144, MATH-147, or MATH-170; one with minimum grade of C-, or achieve sufficient score on placement assessment.

## CS-151 Computer Science II

### 4 Credits

**Lecture:** 3 hours per week, **Lab:** 2 hours per week

**Offering:** Fall and Spring Only, All Years

This course provides continuing experience in problem solving and software design methods. The exploration of recursion is continued and the entire software-design cycle is considered in greater depth. Introduction to abstract data types and fundamental data structures will cover topics: writing code to generate, use, and maintain complex dynamic structures, including linked lists, pointers, stacks, queues, sorts, searches, and trees. Other topics include a continued development of skills in the analysis of algorithms, dynamic memory use, and the use of external files.

**Prerequisites:** CS-150

**Corequisites:** CS-151L

**Pre/Corequisites:** MATH-187

## CS-155 Computer Organization and Assembly Language

### 3 Credits

**Lecture:** 3 hours per week

**Offering:** Spring Only, All Years

This course covers topics including digital logic, machine-level representation of data, and processor architecture covering the ALU, control unit, assembly language, memory organization, addressing methods, I/O and interrupts.

**Prerequisites:** CS-150

**Pre/Corequisites:** MATH-187

## CS-210 Programming Languages

### 3 Credits

**Lecture:** 3 hours per week

**Offering:** Fall Only, All Years

This course develops fundamental concepts of major programming languages, with primary emphasis on language features and their role in designing code and software. Students will study the constructs of programming language design including a conceptual study of procedural, data-flow, functional, and object-oriented languages.

**Prerequisites:** CS-151

## CS-240 Digital Logic

### 4 Credits

**Lecture:** 3 hours per week, **Lab:** 2 hours per week

**Offering:** Spring Only, Even Years

This course includes the following topics: digital logic concepts, logic design, Karnaugh maps, combinational and sequential networks, state tables, state machines, and programmable logic arrays. Laboratory activities use basic lab equipment, logic analyzers, and digital oscilloscopes.

**Prerequisites:** MATH-170 or MATH-187

**Corequisites:** CS-240L

## CS-241 Computer Operating Systems

### 3 Credits

**Lecture:** 3 hours per week

**Offering:** Spring Only, All Years

This course provides an overview of operating systems and operating system principles. It includes sections on concurrency, scheduling and dispatch, memory management, net-centric computing, OS security, and process management. Concurrent programming using threads is also explored.

**Prerequisites:** CS-151, CS-155

**Pre/Corequisites:** CS-228 or CS-270

## CS-270 System Software

### 3 Credits

**Lecture:** 3 hours per week

**Offering:** Fall Only, All Years

This course is designed to provide an introduction to the UNIX operating system and variants (such as Linux) as well as system programming concepts. Programming productivity tools will be introduced such as making, debugging, linking, and loading tools. Shell programming and scripting languages will also be used. System programming tools include process management and interprocess communication, exception handling, network concepts, and network programming.

**Prerequisites:** CS-151

**CS-290 Computer Science Internship**

**1-6 Credits**

**Lecture:** 1 hour per week

**Offering:** Fall, Spring, and Summer, All Years

An off-campus experience directed by an on-site supervisor, but overseen by a faculty member designated to provide the student with an opportunity to observe and/or participate in a job-related activity that falls within the student's field of study. Six credits maximum may be applied toward graduation. Instructor permission required.

**CS-298 Computer Science Practicum**

**1-8 Credits**

**Lecture:** 1 hour per week

**Offering:** Fall, Spring, and Summer, All Years

An out-of-classroom experience designed to give learned in academic course work to specific community-related or employment-related situations. Practicums are overseen by a faculty member. Eight credits maximum may be applied toward graduation. Instructor permission required.