COMPUTER SCIENCE (CSCI)

CSCI 110 Fundamentals of Computer Science
3.5 Units (Degree Applicable, CSU, UC)
UC Credit Limitation
Lecture: 54 Lab: 27
Advisory: *Eligibility for ENGL 1A or ENGL 1AH or ENGL 1AM or AMLA 1A*

Computer hardware and software. General computer organization and information representation. Binary and hexadecimal number systems. Algorithm design and problem-solving techniques. Introduction to programming using a high level language such as C++, Java, or Python.

CSCI 140 C++ Language and Object Development
4 Units (Degree Applicable, CSU, UC, C-ID #: COMP 122)
Lecture: 54 Lab: 54
Prerequisite: *CSCI 110*

For computer science, mathematics, engineering and other science students. C++ programming and object-oriented paradigm. Control structures, functions, arrays, pointers and strings, classes and data abstraction, C++ object programming, operator overloading, inheritance, virtual functions and polymorphism, stream input and output, templates, exception handling, file processing. Data structures in C++, string processing and recursion.

CSCI 145 Java Language and Object Oriented Programming
4 Units (Degree Applicable, CSU, UC)
Lecture: 54 Lab: 54
Prerequisite: *CSCI 110*

Java language and object-oriented programming (OOP) with Java as well as general concepts and techniques of computer programming. Topics include: Java expressions, flow control, methods and program structure, Java classes, overloading, object references, inheritance, Java library packages, exceptions, file input/output (I/O), applets, graphical user interface (GUI), and event handling. A course for computer science, engineering, mathematics, and other science students.

CSCI 150 Assembly Language/Machine Architecture
3.5 Units (Degree Applicable, CSU, UC)
Lecture: 54 Lab: 27
Prerequisite: *CSCI 110*
Advisory: *CSCI 140 or CSCI 145*

Organization and operation of real computer systems at the assembly language level using the Intel 80x86 family of processors; mapping statements and constructs in a high-level language onto sequences of machine instructions; internal representations of simple data types and structures; numerical computation, noting various data representation errors and potential procedural errors; investigation of basic principles of operating systems; and programming language translation process.

CSCI 190 Discrete Mathematics Applied to Computer Science
4 Units (Degree Applicable, CSU, UC)
Lecture: 72
Prerequisite: *MATH 130 and CSCI 110*

A study of set theory, propositional and predicate calculus, modular arithmetic, counting techniques, combinatorics, mathematical induction, recursion, binary search trees, graphs, and finite probability. For students in computers science, engineering, mathematics and other sciences.

CSCI 220 Data Structures I
3.5 Units (Degree Applicable, CSU, UC)
Lecture: 54 Lab: 27
Prerequisite: *CSCI 140 or CSCI 145*

Abstract data types and running time analysis tools. Linear data structures including sets, stacks, queues, and linked lists. Trees, binary search trees, heaps, and priority queues. Many procedures are discussed using an algorithmic language and selected problems are programmed in a higher level language.

CSCI 230 Data Structures II
3.5 Units (Degree Applicable, CSU, UC)
Lecture: 54 Lab: 27
Prerequisite: *CSCI 220*

Basic searching/sorting algorithms, hashing, graphs, memory/disk management, indexing, B-trees, advanced tree structures and analysis.

CSCI 240 Data Structures and Algorithms
5 Units (Degree Applicable, CSU, UC)
Lecture: 72 Lab: 54
Prerequisite: *CSCI 140 or CSCI 145*

Abstract data types and algorithm analysis and design. Linear data structures including stacks, queues, vectors, and lists. Non-linear data structures including trees, binary search trees, heaps, priority queues, and graphs. Searching, sorting, and hash tables. Design patterns including divide-and-conquer, greedy method, and dynamic programming. Memory and disk management techniques.