SANTA ANA COLLEGE

Student Learning Outcomes

Computer Science

CSCI 110 SLO

1. Students will be able to use and differentiate between basic concepts of computer hardware and software.

2. Students will be able to use data representation for the fundamental data types and perform conversions between binary-hexadecimal-decimal representations.

3. Students will be able to read, understand and trace the execution of programs written in C language.

4. For a given algorithm students will be able to write the C code using a modular approach.

CSCI 110 CMO

1. Define computer terminology.

2. Describe various data representations.

3. Demonstrate number system conversion to and from binary, decimal and hexadecimal.

4. Discuss fundamental units of digital computers.

5. Describe instruction set, computer organization and operating system features.

6. Analyze and design efficient algorithms for problem solving.

7. Utilize text editors, compilers and IDEs.

8. Utilize appropriate data types and structures.

9. Write, organize and assemble program documentation.

10. Create correct code, and debug simple errors in one of the higher level languages (C, C++ or Java.)

CSCI 140 SLO

1. Students will be able to analyze problems and design algorithms in pseudo code.
2. Students will be able to read, understand and trace the execution of programs written in C++ language.
3. Students will be able to use given classes and virtual functions in a class hierarchy to create new derived classes and the code that uses them.
4. For a given algorithm students will be able to write modular C++ code using classes in an OOP approach.

CSCI 140 CMO

1. Analyze problems and design appropriate algorithms.
2. Code algorithms into the C++ language.
3. Recognize and produce proper C++ syntax.
4. Use correct data type and data structures, including objects, linked lists, stacks, and queues.
5. Utilize recursion, iteration, arrays, pointer arithmetic.
6. Demonstrate the paradigm of object oriented programming.
7. Write, organize and assemble program documentation.
8. Utilize encapsulation, overloading, inheritance and polymorphism.
9. Utilize data abstraction, separate program interface and implementation.
10. Develop standards for comparing the efficiency of various algorithms.
11. Demonstrate debugging techniques.

SANTA ANA COLLEGE

Student Learning Outcomes

CSCI 150 CMO

1. Explain different number systems and manipulate bits and bytes.
2. Identify the components of a computer and the organization of those components.
3. Use assembly language instructions to write programs.
4. Map statements and constructs in a high-level language into a sequence of machine instructions.
5. Construct internal representations of simple data types.
6. Identify the basic principles of the operating system.
7. Utilize procedures in assembly programs.
8. Analyze disk operations and compare different file systems.
9. Write programs in assembly language.

CSCI 170 SLO

1. Students will be able do basic UNIX OS administration tasks, including account management.
2. Students will be able to use the Unix file system.
3. Students will be able to perform basic UNIX networking tasks including setting up a LAN using NIS.
4. Students will be able to use Unix programming tools: compilers, Make utility, debugger, profiler, version control.
5. Students will be able to read-understand-write short scripts in a Unix shell.

CSCI 170 CMO

1. Describe the function performed by an operating system.
2. Utilize vi and emacs text editors.
3. Write scripts for shell programming in UNIX.
4. Discuss features of UNIX implementations and compare to other operating systems.
5. Manage system administration on SOLARIS and LINUX boxes.
6. Utilize file systems under UNIX.
7. Utilize process management under UNIX.
8. Configure user shells and perform basic tasks as "root" on the system.
9. Explain networking basics, routing, TCP/IP and DNS.
10. Utilize compilers for C/C++, Java under UNIX.

CSCI 190 SLO

1. Students will be able to use truth table for propositional calculus.
2. Students will be able to use math induction and recursive definitions and algorithms.
3.
4. Students will be able to understand the terminology of finite graphs and trees and use the basic algorithms for traversal, shortest path, graph coloring.
5. Students will be able to use basic counting techniques, combinatorics concepts and binomial coefficients.

CSCI 190 CMO

1. Utilize the appropriate mathematical tool in algorithm design.

2. Define problems in mathematical terms using the language of sets, logic, arithmetic, combinatorics.
3. Compose proofs using truth tables or predicate calculus.
4. Develop algorithms using recursion.
5. Utilize modular arithmetic and integer arithmetic in problem solving with computers.
6. Demonstrate elementary counting techniques.
7. Solve problems using mathematical induction.
8. Utilize the language of graphs in problem solving and algorithm design.
9. Classify problems according to mathematical aspect that is relevant to it.
10. Utilize discrete probability for practical problems.

CSCI 210 SLO

1. Students will be able to use Boolean algebra for algebraic simplification.
2. Students will be able to use truth tables, maps, and tabular reduction methods in combinational network design.
3. Students will be able to use state tables and diagrams in sequential network design.
4. Students will be able to differentiate between combinational and sequential logic networks.

CSCI 220 SLO

1. Students will be able to analyze problems and select the appropriate data structure.
2. Students will be able to estimate running time given an algorithm.

3. Students will be able to implement and use linear data structures including sets, stacks, queues, and lists.

4. Students will be able to implement and use trees including binary tree, binary search trees, and heaps.

CSCI 220 CMO

1. Analyze problems and select the appropriate data structure.

2. Design the most efficient data structure for solving a problem.

3. Implement the data structure through effective C++/Java code.

4. Utilize effective search, insertion and deletion algorithms.

5. Demonstrate effective debugging techniques.

6. Write and organize documentation for data structures.
7.
8. Estimate running time for the algorithm studied in class or new algorithms.

CSCI 230 SLO

1. Students will be able to implement efficient searching techniques including hash tables and skip lists.

2. Students will be able to implement and analyze running time for various sorting algorithms.

3. Students will be able to represent graphs and implement well-known graph algorithms.

4. Students will be able to differentiate the costs between memory access and disk access.

SANTA ANA COLLEGE

Student Learning Outcomes

CSCI 230 CMO

1. Analyze algorithms and select the most efficient one to solve a problem.
2. Implement sorting algorithms.
3. Implement hashing algorithms.
4. Use self-organizing lists in problem solving.
5.
6. Understand and implement graph algorithms.
7. Estimate running time for sort, search, and graph algorithms.
8. Identify main memory access and disk access costs.
9. Utilize object-oriented techniques in design of data structures and algorithms.

Source: Mt. Sac College