MSUM/M State Programming Sequence Proposal

Computer Science 0: Principles of Computer Science

DESCRIPTION:

Credits: 3
Lecture Hours/Week: 3
Prerequisites: College Algebra or equivalent

“Introduction to the concepts of problem solving, algorithm development, and structured programming.”

COURSE EFFECTIVE DATES: Fall 2024 through present

OUTLINE OF MAJOR CONTENT AREAS

1. Creative Development
2. Data
3. Algorithms and Programming
4. Computer Systems and Networks
5. Impact of Computing

Learning Outcomes (General)

1. Articulate how programs are developed.
2. Describe how data is represented, abstracted, and communicated.
3. Understand and describe the elements behind how programs work.
4. Explain the relationship behind hardware, software, and networks, including the concepts of assembly language and the Von Neumann model.
5. Understand some current issues about computers and society, potentially including, but not limited to security, cyber-attacks, AI, cloud computing, and IoT.
6. Understand how to use technology securely.
7. Demonstrate the ability to navigate, modify, and assess file systems.
8. Demonstrate the ability to navigate, modify, and assess file systems.

Computer Science 1: Introduction to Computers and Programming 1

COURSE DESCRIPTION

Credits: 4
Prerequisites: CS0 or permission of instructor

“Introduction to problem solving, basic algorithm development, data abstraction, structured and object-oriented programming in a high-level language”

COURSE EFFECTIVE DATES: Fall 2024 through present

OUTLINE OF MAJOR CONTENT AREAS

1. Problem Solving and Basic Algorithm Development
2. Computer program development and execution
3. Object-Oriented Programming Paradigm

Learning Outcomes (General)

1. Develop solutions to problems.
2. Develop algorithms suitable for translation to a high-level programming language.
3. Demonstrate the ability to write computer programs in a high-level programming language.
4. Describe the general principles of Object-Oriented Programming, including composition and inheritance (derivation).
5. Describe basic Abstract Data Types and their uses.
6. Use programming language documentation to implement a problem solution.
7. Demonstrate the ability to navigate, modify, and assess file systems.
8. Demonstrate the ability to navigate, modify, and assess file systems.

Data Structures

COURSE DESCRIPTION

Credits: 4
Prerequisites: CS1 or equivalent AND Concepts from Discrete Mathematics or equivalent

“A survey of data representation, data structure implementation, and algorithms using those data structures. Data structures that are reviewed include lists, strings, trees, graphs, and records.”

for graphs, trees, strings, sorting and searching. 3 Cr. DEMAND.

COURSE EFFECTIVE DATES: Fall 2024 through present

OUTLINE OF MAJOR CONTENT AREAS

1. Data Structures
2. Algorithm analysis
3. Recursion
4. Data representation

Learning Outcomes (General)

1. Design and implement user defined data structures.
2. Use a programming language to investigate the application of structures, including lists, stacks, queues, dictionaries, and binary search trees to problem solutions.
3. Use a programming language to investigate the behavior and implementation of classical search techniques.
4. Implement recursive and iterative algorithms over appropriate data structures (e.g., tree traversal, pre-order, and post-order).
5. Analyze recursive and iterative algorithms for space and time complexity.
6. Identify appropriate use of and implement statically allocated data structures.
7. Identify appropriate use of and implement dynamically allocated data structures.

8. Develop and implement a dictionary data type incorporating hashing algorithms and collision avoidance.
9. Write programs that use each of the following data structures: arrays, records/structs, strings, linked lists, stacks, and queues.
10. Implement algorithms and underlying data structures utilizing standard object-oriented design principles.