STRANDS AND STANDARDS
ALGORITHMS & DATA STRUCTURES

Learning that works for Utah
CTE®

Course Description

This course builds on the object-oriented programming principles taught in Computer Programming 1, 2, and Advanced. A solid understanding of these concepts is assumed and required in this course. This course presents the ideas, tools, structure, syntax, libraries, and object-oriented design techniques for developing well-formed programs using data structures. Student's study and strengthen their concepts such as problem solving, program structure, classes, methods, data types, control constructs, file and console I/O. Students will also learn other important principles in designing object-oriented programs containing data structures. Students will design and use common data structures including arrays, hash tables, stacks, queues, linked lists, binary trees, multiway trees, graphs. Students will define and use common algorithms including traversals, searching, sorting, compression, and paths. Students will write several programs that demonstrate their understanding of these concepts using an appropriate programming language.

Intended Grade Level	11-12
Units of Credit	1.0
Core Code	35.02.00.00.035
Concurrent Enrollment Core Code	35.02.00.13.035
Prerequisite	Computer Programming Advanced
Skill Certification Test Number	845
Test Weight	1.0
License Area of Concentration	CTE and/or Secondary Education 6-12
Required Endorsement(s)	
Endorsement 1	Programming & Software Development

STRAND 1

Students will increase their ability to do object-oriented design and implement object- oriented programs using a programming language.

Standard 1

Analyze and perform algorithmic analysis of data operations including:

- Insert
- Remove
- Search
- Sort

STRAND 2

The students will apply good object-oriented design and analysis methods to develop programs and refine their programming skills.

Standard 1

Using an integrated development environment (IDE) and appropriate design procedures, construct reasonably complex programs.

- Understand the concept of pointers and reference variables
- Use inheritance and polymorphism
- Create and use function templates or generics
- Create and use class templates or generics
- Use standard exception handling techniques
- Use recursion
- Use appropriate built-in structures and algorithms from the language in use, including but not limited to:
  - C++ Standard Template Library
  - Java Collections Framework
  - Python Standard libs
  - C# Collections
  - Swift Standard Library

Standard 2

Use appropriate design procedures.

- Design and use simple and complex data structures to solve sophisticated problems
- Provide reporting from data

Standard 3

List the Big-O notation categories and match them to a program with its Big-O notation.

STRAND 3

The students will be able to test and document their programs.

Standard 1

Test and debug programs to assure their quality and usability.

- Use language specific unit testing libraries:
  - JUnit
  - NUnit

- PyUnit
- XCTest

Standard 2

Document programs for understandability and maintainability.

- By providing in-line comments
- By standardized class and file headers
- By using elements of good programming style including multiline comments to provide documentation
  - C++:
    - /** to */
  - C#: Documentation Comments
    - ///
  - Java: Javadoc Comments
    - /** to */
  - Python: Docstrings
    - “”” to “””
  - Swift
    - Markup Documentation

STRAND 4

The students will demonstrate an understanding of different linear and non-linear data structures.

Standard 1

Use arrays.

Standard 2

Use dynamic linear structures.

- Linked Lists
- Stacks
- Queues

Standard 3

Use binary trees and multiway (n-ary) trees.

Standard 4

Use graphs.

Standard 5

Use hash tables.

Performance Skills

Students will list the benefits and issues of using arrays, linked lists, etc. in relation to computer architecture and performance.

STRAND 5

Students will explore and implement multiple algorithms including at least the following:

- Sorting
- Searching
- Traversal

- Compression
- Dynamic Programming
- Shortest Path

STRAND 6

The students will use appropriate data structures.

Standard 1

Discuss the basic principles of many software data structures, including efficiencies and tradeoffs including size, speed, complexity.

Standard 2

Implement and use several data structures in programs

- Using large data sets (greater than 10K entries)
- Read and write to a file
- Timing of code execution

Overall Performance Skills

Create applications/programs highlighting proper use of linear and non-linear data structures and demonstrating correct use of standard algorithms including:

- Using a stack to process postfix notation
- Timing execution of events in a priority queue
- Traversing a graph of nodes representing a real-world location

Workplace Skills

Workplace Skills taught:

- Communication
- Problem Solving
- Teamwork
- Critical Thinking
- Dependability
- Accountability
- Legal requirements / expectations

Skill Certification Test Points by Strand

Test Name	Test #	Number of Test Points by Strand										Total Points	Total Questions
		1	2	3	4	5	6	7	8	9	10		
Alg & Data Structures	845	2	5	2	9	9	4					31	30