STRANDS AND STANDARDS
ALGORITHMS AND DATA STRUCTURES

Course Description

This course builds on the object-oriented programming principles taught in Computer Programming 1, 2, and 3. 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. Students 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 including: C++, C#, Java, Python, and Swift.

CTE® Learning that works for Utah

ADA Compliant: July 2018

Intended Grade Level	10-12
Units of Credit	1.0
Core Code	35.02.00.00.037
Concurrent Enrollment Core Code	35.02.00.13.037
Prerequisite	Computer Programming 1, 2, and 3 or equivalent
Skill Certification Test Number	TBD
Test Weight	0.0
License Type	CTE and/or Secondary Education 6-12
Required Endorsement(s)	
Endorsement 1	Computer Science – Level 2

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

Work Place Skills

Communication and Problem Solving.

Standard 2

Use appropriate design procedures

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

Work Place Skills

Critical thinking

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.
  - C++:
    - /** to */
  - C#: Documentation Comments
    - ///
  - Java: Javadoc Comments
    - /** to */
  - Python: Docstrings
    - """ to """
  - Swift:
    - Markup Documentation

Work Place Skills

Problem Solving, Teamwork, Critical Thinking, Dependability, Accountability

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 hastables.

STRAND 5

The students will explore multiple algorithms

- 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

Work Place Skills

Communication, Problem Solving, Teamwork, Critical Thinking, Dependability, and Accountability.

Performance Skills

Create an application/program highlighting proper linear and non-linear structures, stacks to evaluate postfix expressions and to convert expressions from infix to postfix and use queues for basic timing simulations.

Skill Certificate Test Points by Strand

Coming Soon!!