Algorithms and Design Patterns 2023-2024 Catal

[ARCHIVED CATALOG]

SDEV 268 - Algorithms and Design Patterns

PREREQUISITES: (SDEV 200 - Software Development using Jav OR SDEV 220 - Software Development Using Python OR SDEV 230 - Software Development using C++ OR SDEV 240 - Software Development Using C# OR CSCI 102 - Computer Science II OR CSCI 201 - Computer Science II).
PREREQUISITES/COREQUISITE: MATH 136 - College Algebra or Higher or demonstrated competency.

PROGRAM: Software Development
CREDIT HOURS MIN: 3
LECTURE HOURS MIN: 2
LAB HOURS MIN: 2
DATE OF LAST REVISION: Fall, 2020

The course is an introduction to the analysis and design of computer software using algorithms and design patterns. Topics will include concepts of algorithm complexity, and various algorithmic design patterns. The course introduces the basics of computational complexity analysis and various algorithms and design pattern paradigms. Students are provided a solid foundation in patterns and algorithms to deal with a wide variety of computational problems. Students will be familiarized with the most common algorithms, patterns, and data structures.

MAJOR COURSE LEARNING OBJECTIVES: Upon successful completion of this course the student will be expected to:

1. Demonstrate with proficiency procedural and object-oriented programming concepts.
2. Demonstrate basic algorithmic complexity analysis.
3. Analyze a problem and evaluate algorithms appropriate for its solution.
4. Compare and analyze different algorithms and their usage.
5. Design and implement an algorithm in software to meet desired needs.
6. Identify a variety of real-world problems in software development solvable using various design patterns.
7. Identify and explain how various patterns can be applied to the construction of software systems.
8. Identify suitable design patterns to address a given application design problem.
9. Design, develop, and test programs that implement various design patterns.
10. Analyze the relationship between design principles, design patterns, and anti- patterns.
11. Investigate security requirements associated with algorithm and pattern usage.

COURSE CONTENT: Topical areas of study include -

- Creational Design Patterns
- Structural Design Patterns
- Behavioral Design Patterns
- Anti-patterns
- Refactoring
- UML
- Documentation
- Teamwork and communication
- Data Structures
- Dynamic Programming
- Algorithms

- Algorithm Analysis
- Collaboration Tools
- Secure Programming
- Standard Libraries
- Big O, Little O, Omega and Theta
- Recursive vs Iterative Solutions
- Peer Reviews
- Objects/Object Factories

Course Addendum - Syllabus (Click to expand)