Computing Logic

2023-2024 Catal

[ARCHIVED CATALOG]

SDEV 120 - Computing Logic

PREREQUISITES: Demonstrated readiness for college-level English and Demonstrated readiness in STEM MATH - Route 2 or QUANT MATH.
PROGRAM: Software Development
CREDIT HOURS MIN: 3
LECTURE HOURS MIN: 3
DATE OF LAST REVISION: Fall, 2020

Introduces the student to algorithms, logic development and flowcharting as tools used to document computer logic. Students will study math concepts and the importance to computer development. Included areas of study are base numbering systems, truth tables, logic and relational operators. Other concepts covered are order of precedence, decision trees, security, different types of language approaches, and scripting. Students will practice skills such as listening, team building, work ethic, communications, documentation, and adaptability. Concepts will be demonstrated using basic scripting and simple programming code.

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

1. Identify the standard documentation tools of displaying algorithms such as pseudocode, flowchart symbols and UML.
2. Describe the basic logical structures, file handling, matrices, and arrays to program algorithms.
3. Apply truth tables, Boolean logic, control structures, logical and relational operators to program algorithms.
4. Use set theory and logic gate theory to develop program algorithms.
5. Use puzzles and games to enhance problem-solving skills.
6. Compare key techniques to visually represent data such as graphs, charts and tables.
7. Express algorithms using tools such as data flow diagrams, flowcharts, use case diagrams, activity diagrams, and state tables.
8. Apply critical thinking and problem solving methodologies.
9. Apply base numbering systems techniques to convert numeric data to any base numbering format, including binary, decimal and hexadecimal.
10. Identify the uses of various programming and scripting languages in computer systems.
11. Develop a simple program and/or script.
12. Compare and contrast software development methodologies as it pertains to software development and problem solving.
13. Discuss the concepts and justifications for using secure design techniques.
14. Describe the different methods for encoding data such as BCD, 1's complement, 2's complement, ASCII and Unicode.
15. Employ team building, work ethic, and communication skills.
16. Describe the components of a computer architecture.

COURSE CONTENT: Topical areas of study include -

- Algorithms and Algorithm Development
- Logic Development
- Math Concepts

- Numbering Systems: Binary, Decimal, Hexadecimal
- Basic Programming Control Structures
- Flowcharting and Pseudocode
- Relational and Logical Operators
- Order of Precedence
- Truth Tables and Decision Trees
- Data Validation
- Scripting
- Work Ethic
- Software Development Life Cycle
- Deductive and Inductive Logic
- Set Theory
- Logic Structures
- Fuzzy Logic and Ladder Logic
- Logic Gate Theory
- Puzzles and Games
- Graphing
- Unified Modeling Language
- Prototyping
- Agile/Iterative
- Von Neumann Machine

Course Addendum - Syllabus (Click to expand)