Course number and name:	**CS 07210: Foundations of Computer Science**
Credits and contact hours:	3 credits. / 3 contact hours
Faculty Coordinator:	Jennifer Kay
Text book, title, author, and year:	Discrete Structures, Logic and Computability, 3rd/4th Edition, James L. Hein, 2009/2017.

Specific course information

Catalog description: This course provides an introduction to the theoretical foundations of computer science, including finite automata, context-free grammars, Turing machines, and formal logic.

Prerequisites: (MATH 03160 Discrete Structures or
MATH 03150 Discrete Mathematics) **and**

(CS 01102 Introduction To Programming or
CS 04103 Computer Science and Programming or
CS 01104 Introduction to Scientific Programming or
CS 04113 Introduction To Object Oriented Programming)

Type of Course: ☒ Required ☐ Elective ☐ Selected Elective

Specific goals for the course

1. **finite automata.** Students have demonstrated understanding of abstract machines including DFAs, NFAs, PDAs, and Turing Machines, and what can be computed with them. Students have produced the appropriate graphical and formal representations of a finite automaton given an English-language description of a language.

    - ABET (a) An ability to apply knowledge of computing and mathematics appropriate to the program's student outcomes and to the discipline

2. **formal languages – overview**. Students have understood the concept of a formal language, how a single language may be specified in multiple ways (e.g. as a set, as a regular expression, finite automaton, grammar).

    - ABET (a) An ability to apply knowledge of computing and mathematics appropriate to the program's student outcomes and to the discipline

3. **formal languages - transforming representations**. Students have transformed between different representations of languages, e.g., regular expressions into NFAs and NFAs into regular expressions.

    - ABET (a) An ability to apply knowledge of computing and mathematics appropriate to the program's student outcomes and to the discipline

4. **formal logic – logic formulae.** Students have taken English-language statements and converted them into formal propositional and predicate logic formulae.

    - ABET (a) An ability to apply knowledge of computing and mathematics appropriate to the program's student outcomes and to the discipline

5. **formal logic – proofs.** Students have constructed formal proofs in propositional logic

    - ABET (a) An ability to apply knowledge of computing and mathematics appropriate to the program's student outcomes and to the discipline

6. **hierarchy of formal languages.** Students have demonstrated understanding of the differences between regular, context free, and non-context-free languages. Given an English-language description of a language, students have given a formal specification for that language and determined if it is regular and/or context-free.

    - ABET (a) An ability to apply knowledge of computing and mathematics appropriate to the program's student outcomes and to the discipline

Required list of topics to be covered

1. Strings & languages
2. Regular expressions
3. DFAs
4. NFAs
5. Transforming NFAs into regular expressions
6. Transforming regular expressions into NFAs
7. PDAs
8. Turing Machines and Turing computability
9. Computability and non-computability
10. Propositional logic
11. Formal proofs in propositional logic
12. Predicate logic
13. Deterministic and non-deterministic finite automata
14. Formal language theory

Optional list of topics that could be covered

1. Regular grammars
2. Context-Free grammars
3. Mealy and Moore Machines

4. Alternate models for PDAs (e.g. accept on empty stack if accept on final state was initially taught).
5. Alternate models for Turing machines (e.g. multi-tape if single tape was taught)
6. The Halting Problem
7. Chomsky Hierarchy
8. Church-Turing Thesis
9. Formal proofs that a language is not regular (and the pumping lemma for regular languages
10. Formal proofs that a language is not context-free (and the pumping lemma for context-free languages
11. Formal proofs in predicate logic
12. Boolean algebra
13. Congruences and public-key cryptography