

CSC 131: Computational Thinking

Catalog Description:

Solving problems using computation and implementing solutions in a high-level programming language. Introduction to problem analysis, solution design, data structures, and algorithms. 4(3-2) F, S

Prerequisites for this course: "C" or better in CSC 130 and eligible for MTH 261

This course is a prerequisite for: CSC 232, 335, 365 and 320.

Required Text:

Spring 2012—Spring 2014 (Python 2)

Fundamentals of Python: From First Programs through Data Structures. By: Kenneth A. Lambert. ISBN-10: 1423902181.

Fall 2014 (Python 3)

Fundamentals of Python: First Programs. By: Kenneth A. Lambert. ISBN-13: 978-1111822705.

Fundamentals of Python: Data Structures. By: Kenneth A. Lambert. ISBN-13: 978-1285752006.

Major Topics (including information for course sequence or transition)

1. Variable scope and parameter passing
2. Higher order functions
3. Design with classes
4. Inheritance
5. Graphical user interfaces
6. Event handling
7. Searching and sorting
8. Collections, arrays, and linked structures
9. Stacks and queues
10. Recursion

Student Outcomes Assessed in CSC 131

- A. Students will attain an ability to apply knowledge of computing and mathematics appropriate to the discipline
- B. Students will attain an ability to analyze a problem, and identify and define the computing requirements appropriate to its solution
- C. Students will attain an ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs

CAC Characteristics Enabled But Not Assessed in CSC 131

- I. Students will attain an ability to use current techniques, skills, and tools necessary for computing practice
- K. Students will attain an ability to apply design and development principles in the construction of software systems of varying complexity

Table 1. Student Outcomes assessed by CSC 131

CSC 131 Student Outcomes	CSC 131 Performance Indicators	CSC 131 Assessment Goals
CSC 131 contributes to SO A: Students will attain an ability to apply knowledge of computing and mathematics appropriate to the discipline	PI 131-1a: Simple big-O analysis PI 131-1b: Selection sort vs. insertion sort PI 131-1c: Write code to manipulate a singly linked structure PI 131-1d: Convert an infix mathematical expression to postfix PI 131-1e: Find the value of a postfix mathematical expression	PI 131-1a: $\geq 60\%$ correct PI 131-1b: $\geq 60\%$ correct PI 131-1c: $\geq 60\%$ correct PI 131-1d: $\geq 60\%$ correct PI 131-1e: $\geq 60\%$ correct
CSC 131 contributes to SO B: Students will attain an ability to analyze a problem, and identify and define the computing requirements appropriate to its solution	PI 131-2: Random walk program	PI 131-2: $\geq 60\%$ correct
CSC 131 contributes to SO C: Students will attain an ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs	PI 131-3a: Program to implement a class with operator overloading PI 131-3b: GUI to compute BMI	PI 131-2: $\geq 70\%$ correct PI 131-3: $\geq 70\%$ correct

Table 2. CAC Characteristics enabled by CSC 131

CSC 131 CAC Characteristics	CSC 131 Characteristics Enablers
CSC 131 enables Characteristic I: Students will attain an ability to use current techniques, skills, and tools necessary for computing practice	CE 131-1: Students write programs that employ different techniques and skills. Most students use IDLE as their computing environment in this course.
CSC 131 enables Characteristic K: Students will attain an ability to apply design and development principles in the construction of software systems of varying complexity	CE 131-2: Students learn the model-view and model-view-controller design patterns.