The University of Georgia

Department of Computer Science

Course Information Sheet

CSCI 1360

Foundations for Informatics and Data Analytics

Brief Course Description
(50-words or less)

Informatics, or "data science," are rapidly becoming essential skills for scientists across fields; in addition to field-specific specializations, researchers require knowledge of and experience with quantitative analytical techniques for extracting knowledge from raw data.

Extended Course Description / Comments

Use this section to put additional information that's relevant to whom this course is targeting

This course aims to provide an introduction to concepts in scientific programming and data science using the Python language. Students are given hands-on opportunities to learn techniques applicable to quantitative analyses across a broad range of fields. These core techniques involve formulating solutions in terms of their inputs and outputs (functional programming), repeated operations (loops), branching operations (conditionals), different methods of organizing data (data structures), how to implement an optimal problem-solving strategy (algorithm design), and methods for visualizing and interpreting results.

Pre-Requisites and/or Co-Requisites

MATH 1113
Precalculus

Approved Textbooks
(If more than one, course text used during a semester is at the discretion of the instructor)

Author(s): Zed Shaw
Title: Learn Python the Hard Way
Edition: 3rd Ed., 2013
ISBN-13: 978-0321884916

Specific Learning Outcomes (Performance Indicators)

These are a (non-exhaustive) list of specific, measurable outcomes, as they relate to the course & program objectives.

These learning outcomes should avoid using ambiguous language such as "understand" or "familiar".

Performance indicators must include an action verb (indentifying the depth to which students should demonstrate performance), and the content referent that is the focus of the instruction (from ABET)

Target number 5 - 10

This course introduces students to concepts in scientific programming and informatics. At the end of the semester, all students will be able to do the following:

1. Frame scientific experiments in terms of their inputs and outputs.
2. Formulate algorithms in terms of conditionals, loops, functions, return values, data structures, and existing Python APIs.
3. Write a program or package to implement automated analysis of data.
4. Process data of varying types, such as text or images.
5. Render appropriate visualizations of analysis results, and interpret these visualizations.

Relationship Between Course Outcomes and Learning Outcomes

		Program Outcomes										
		A	b	c	d	e	f	g	h	i	j	k
Learning Outcomes	1	•	•									•
	2	•								•		•
	3	•	•	•						•	•	•
	4	•								•	•	
	5	•					•			•		

Program Outcomes

(These are ABET-specified and should not be changed)

a. An ability to apply knowledge of computing and mathematics appropriate to the discipline.
b. An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution.
c. An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs.
d. An ability to function effectively on teams to accomplish a common goal.
e. An understanding of professional, ethical, legal, security and social issues and responsibilities.
f. An ability to communicate effectively with a range of audiences.
g. An ability to analyze the local and global impact of computing on individuals, organizations, and society.
h. Recognition of the need for and an ability to engage in continuing professional development.
i. An ability to use current techniques, skills, and tools necessary for computing practice.
j. An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices.
k. An ability to apply design and development principles in the construction of software systems of varying complexity.

Major Topics Covered
(Approximate Course Hours)

3 credit hours = 37.5 contact hours
4 credit hours = 50 contact hours

Note: Exams count as a major topic covered

Introduction to informatics (2.5-hours)
"Hello World" and Python Variable types (2.5-hours)
Loops, Conditionals, and Control Flow (5-hours)
Data Structures: lists, arrays, dictionaries, sets, matrices (7.5-hours)
Functions and Functional Programming (5-hours)
Vectorized Programming (2.5-hours)
Data Formats, Transformations, and Preprocessing (5-hours)
Algorithms: regression, classification, clustering (10-hours)
Data Visualization (7.5-hours)
Extending the Python Ecosystem (2.5-hours)

Assessment Plan for this Course

Each time this course is offered, the class is initially informed of the Course Outcomes listed in this document, and they are included in the syllabus. At the end of the semester, an anonymous survey is administered to the class where each student is asked to rate how well the outcome was achieved. The choices provided use a 5-point Likert

scale containing the following options: Strongly agree, Agree, Neither agree or disagree, disagree, and strongly disagree. The results of the anonymous survey are tabulated and results returned to the instructor of the course.

The course instructor takes the results of the survey, combined with sample student responses to homework and final exam questions corresponding to course outcomes, and reports these results to the ABET committee. If necessary, the instructor also writes a recommendation to the ABET committee for better achieving the course outcomes the next time the course is offered.

How Data is Used to Assess Program Outcomes

Each course Learning Outcome, listed above, directly supports one or more of the Program Outcomes, as is listed in "Relationships between Learning Outcomes and Program Outcomes". For CSCI 1360, Program Outcomes (a), (b), (c), (f), (i), (j), and (k) are supported.
Dr. Shannon Quinn

Course Master

Course History