Fall 2018 - CS 100 - Exam 2 Review Suggestions

last modified: 2018-10-30

- CS 100 Exam 2 will be given during class on Friday, November 2.
- You are responsible for material covered in reading assignments, class sessions, and homeworks; but, here's a quick overview of especially important material.
- This exam will be similar in style to Exam 1, although most of the questions will focus on "new" material (material covered since Exam 1). However, concepts from Exam 1 will still be involved -- we have necessarily been building on earlier material.
 - So, while the focus of most of the questions will be:
 - Chapter 9 A Little Categorical Logic
 - Chapter 10 A Little Propositional Logic
 - the in-lecture coverage of Scheme and WeScheme
 - Homeworks 5, 6, 7, and 8
 - ...you still need concepts from Chapters 1, 2, and 3 in the course text and Homeworks 1, 2, 3, and 4 as well.
 - (You are *not* responsible for Monday, October 28's discussion of Chapter 7, on diagramming arguments, for Exam 2; you will be responsible for that material on the Final Exam.)
- You are permitted to bring into the exam a single piece of paper (8.5" by 11") on which you have **handwritten** whatever you wish on one or both sides. This paper must include your name, it must be handwritten by you, and it will **not** be returned.
 - Other than this piece of paper, the exam is closed-note, closed-book, and closed-computer/closed-electronic-devices.
 - You are to work individually on all exams in this course.
- This will be a pencil-and-paper exam. You only need to bring something to write with, and, if you'd like, the page of notes mentioned above.
- Your studying should include careful study of assigned readings, posted examples and notes, and homeworks (and posted solutions) thus far.
- I expect that the exam questions will be a combination of short-answer and multiple choice.
- Based on suggestions from Prof. Deb Pires from UCLA: Because of the research-supported learning potential when students study together and explain concepts to one another, you will receive (a maximum) ***5 POINTS BONUS*** on Exam 2 if you do the following:
 - set up and/or attend an exam study session with at least one other CS 100 student (but it can involve as many other CS 100 students as you would like!)
 - for **YOU** to receive the bonus, **YOU** SEND an e-mail to me, sent on the **same day** as the study session (which needs to take place **before you take Exam 2**) in which you:

- use as the e-mail Subject: CS 100 Exam 2 bonus
- briefly DESCRIBE (a sentence or two is fine) what you covered at the exam study session
- INCLUDE a picture of everyone involved in the study group (or at least yourself and at least one other CS 100 student)
- (EACH person who wants the 5 point bonus should send an e-mail message to me containing this Subject: line AND these TWO parts.)
- Please let me know if you have any questions about this, and I hope it encourages you to study together for Exam 2.

Chapter 9 - A Little Categorical Logic

- You should know what a categorical statement is; you should be able to answer questions about categorical statements.
- You should know the four standard categorical forms:
 - All A are B
 - No A are B
 - Some A are B
 - Some A are not B
- EXPECT IT -- you will need to determine if statements are in standard categorical form.
- You should be able to convert non-standard categorical statements into standard categorical form.
- What is a Venn diagram? How can Venn diagrams be used to depict categorical statements?
- You should be able to look at a Venn diagram, and understand what it means in categorical statement terms; you should be able to answer questions about Venn diagrams.
- Given a Venn diagram with 2 or 3 overlapping circles, what are the sets/classifications in that diagram?
- What does it mean to shade a portion of a Venn diagram? What does it mean to put an X in a portion of a Venn diagram?
- EXPECT IT Drawing and/or identifying appropriate Venn diagrams for categorical statements.
- What is a categorical syllogism, and how is it represented using a Venn diagram?
- EXPECT IT Drawing and/or identifying Venn diagrams to show whether categorical syllogisms are valid or not.
 - Given a categorical syllogism, you should be able to mark up a 3-circle Venn diagram appropriately.
 - And, you should be able to then state whether that argument is VALID or INVALID based on the resulting Venn diagram.
- You could also be asked questions about the process of making Venn diagrams to show whether categorical syllogisms are valid.

Chapter 10 - A Little Propositional Logic

- EXPECT IT -- you will have to write truth tables in propositional logic form.
- You should be familiar with and able to write the truth tables for logical "AND" (using & and ^), logical "OR" (v), logical "NOT" (~), and logical IF-THEN/"IMPLIES" (→). You should also know their truth tables, and be able to create truth tables for statements that use these symbols.
- You should know and be able to follow the propositional logic conventions for truth tables as used in class.
- You should be able to read an argument expressed in propositional logic form; you should be able to convert an argument into propositional logic form, using letters to represent simple statements and AND/OR/NOT/IF-THEN to build compound statements. Write the premises in symbolic form above the conclusion (also in symbolic form), and use the special \therefore symbol to mark the conclusion.
- EXPECT IT -- you will have to write and/or identify arguments in propositional logic form.
- EXPECT IT -- you will have to use truth tables to show whether deductive arguments (that can be written in propositional logic form) are valid or invalid.
 - For a deductive argument given in propositional logic form, what columns would one include?
 - When using a truth table to determine if a deductive argument is valid or invalid, know what it means if a column heading is marked with (*), and what it means if a column heading is marked with (C)
 - When using a truth table to determine if a deductive argument is valid or invalid, know how to build that truth table and fill in all of its needed rows.
 - Once filled out, know which rows you should cross out as part of determining if a deductive argument is valid or invalid,

and know how to determine, based on the rows remaining, whether that means that the deductive argument has been shown to be valid or invalid.

• You should be able to answer questions about the steps involved in using truth table to show whether deductive arguments (that can be written in propositional logic form) are valid or invalid.

In-lecture coverage of Scheme and WeScheme

- EXPECT IT: given a mathematical expression, you will write the expression in the Scheme language.
- In Scheme, what is a simple expression? What is a compound expression? How do you write simple and compound expressions in Scheme?
 - Given an expression, you should be able to tell if it is a syntactically-acceptable Scheme expression.
 - You should be able to tell if a Scheme expression is simple or compound; you should be able to write a simple or compound Scheme expression.
- What is **prefix notation**? How is a simple mathematical expression written in prefix notation?
- What is the purpose of parentheses in the Scheme language? When do you use parentheses?
- Which standard Scheme data types have we discussed and used in class and assignments?

- How can you write a comment in Scheme?
- What are some of the arithmetic operations/functions provided by Scheme? Given a numeric operation in standard "algebraic" form, you should be able to write it as an equivalent Scheme expression.
- What is a **function header**? What is a **function body**?
- Given a function definition, you should be able to tell what part is its header, and what part is its body.
- You should be comfortable with the "Recipe" feature on <u>www.wescheme.org</u>, since you have been using it for all of the functions that you have written
- EXPECT IT: given a "fill-in-the-blank" recipe screen, you should be able to write the correct items in the blanks for defining a Scheme function.