CS 232 – Programming in Python

Exam #1 Study Guide

Date and Time

The first exam will be held in Lab on **Thursday, March 3, 2016 @ 5:00 PM AND 6:00 PM (see below).**

Test Format and Materials Allowed

- The exam will be written to be **50 minutes long**. The exam will be given in **two periods** – students will be assigned a 50-minute period in which to take the exam, either 5:00-5:50 PM or 6:00-6:50 PM. This is to allow for the fact that we have more students in the class than there are seats in the lab, and to allow for some non-functioning PCs in the lab.
- Modifications to the assignment schedule will be considered on a case-by-case basis, but there is no guarantee that requests will be approved – the lab period for Python is 5:00-6:50 PM on Thursdays, and all students are expected to be available during that entire period.
- You will be permitted to being a single 8.5" by 11" (letter-sized or equivalent) sheet of notes (both sides OK) that are hand-written or personally created using a text editor or word processor. Do NOT photocopy material from the text or cut-and-paste graphics into your notes. This sheet of notes is to be turned in with your exam and must include your name at the top. It will be returned to you with the graded exam.
- Other than that, bring only a pencil or pen (pencil is greatly preferred). You will be writing all answers to the questions on the exam onto the exam paper – no electronic submissions of any kind.
- You will be seated in front of a desktop PC in the computer lab in BSS 313. You will be permitted to use that PC **but only under the following conditions**:
  - The **only** application you may use is the Python IDLE application installed on the PC. No web browsers, no email, no other applications – ONLY the Python IDLE application! This will be closely monitored!
  - You may NOT access any pre-existing Python code from your U: drive or from any other source.
  - You will be permitted to use the interactive window and (optionally) create **new, empty** Python module files in which to write, execute, and test Python code.
  - You will be submitting all answers to the exam **on your test paper**. I will not accept Python code in electronic form! The Python application will simply be a way for you to check your work and ensure your Python code runs as you expect it to.
- **This is an experiment** to expand the options that students have to take programming exams in a controlled environment that is **equal and fair** to all students. Requests for unequal levels of access to resources (using one's own laptop, for example) will NOT be granted.
- Students who are entitled to accommodations when taking exams must make such arrangements in advance, in accordance with SDRC and Testing Center rules.

Material Covered

- You are responsible for material covered in class and lab through Week 5 (the week ending February 19). In the textbook, this includes the material from Chapters 1 through 3 as discussed in class or lab or assignments:
  - Variables, Names and Objects
  - Atomic Data Types – int, float, str, bool
  - Built-in Functions and Operations on atomic data types, as discussed in class / lab / assignments
  - Molecular Data Types – lists, tuples, dictionaries, sets
  - Built-in Functions and Operations on molecular data types, as discussed in class / lab / assignments
- Part of Chapter 4. The material in Chapter 4 that will be covered includes:
  - Function Definitions and Calls to Functions
  - Conditionals
  - While and For Loops, and Iterators like range()
  - Break and Continue
  - Comprehensions
  - Positional and Keyword Arguments.
- Things that will NOT be on this exam: Docstrings, Passing functions, Inner functions, Lambda functions, Decorators, Namespace / Scope, and other material in Chapter 4 not yet covered (such as Exceptions and Error Handling).

What You Can Expect on the Exam

- You should expect questions of these types on the exam:
  - Give definitions to vocabulary words and phrases used in Python programming
  - Write expressions in Python that are of the various atomic and molecular data types we've discussed so far in class / lab / assignments
  - Write expressions in Python that meet certain criteria, as expressed in vocabulary covered in class
  - Set values to variables, and use those variables appropriately in Python expressions
  - Given Python code containing syntax errors, fix the errors to write correct Python code
  - The use of predicate (boolean) functions and type casting functions (such as str() and int())Given a Python function definition, modify it to perform a related task or to expand its functionality
  - Use built-in functions, methods, and operations on standard Python atomic and molecular data types, as discussed in class / lab / assignments, to perform a requested task
  - Define Python functions that meet the specifications given in a problem
  - Write expressions in Python that call the functions mentioned above to perform certain tasks