CS 5010 – Exam 2 – Study Guide

Topics that may be covered on Exam 2 *(note, this is **not** an exhaustive list):*

Exam 1 topics may show up again in the context of Exam 2. Focus will be on Modules 4-6. Reference Exam 1 Study Guide for topics covered.

1. Python Coding / CS Programming Concepts (**General**) –
   Should any Python coding/reading of code or related CS programming topics be present on the exam, follow these general guidelines
   a. Be able to read / write Python code
   b. Be able to show the output of a piece of code
   c. Be able to compare two pieces of code
   d. Describe the (many!) benefits of applying object-oriented principles
   e. Describe what the major data structures are and their properties (and scenarios in which they would be best suited – e.g. lists, dictionaries, and tuples)
   f. Decide on an appropriate data structure(s) to use given the scenario
   g. Correctly implement the appropriate data structure(s) for the given scenario and manipulate data within this structure correctly and appropriately
   h. Describe the major functions used and their properties (and scenarios in which they would be best suited)
   i. Be familiar with searching (*linear* and *binary*) and sorting algorithms. You *will* be expected to demonstrate knowledge of the algorithms by answering questions about them, or deciding which one is most appropriate. Built-in sorting algorithms will not be covered on this exam.
   j. A high-level question might be asked on NumPy & Pandas (*no coding* will be involved, however) (Related: querying data: high-level questions; no code.)
   k. Explain some of the elements that results in "good quality" code

2. Testing & Debugging
   a. Explain what testing is
   b. Why is testing important?
   c. Explain the difference between white-box and black-box testing
   d. Best practices
   e. Test-driven development – including benefits
   f. Be able to write a unit test (other class code will be provided), be familiar with at least the top 3 assert statements (assertEqual, assertTrue, assertFalse)
   g. When writing unit tests, remember there are *no* return statements!
   h. Amount of *time* spend on testing vs. writing main code
      i. You'll spending more time writing tests
   i. Amount of *code written* for testing vs. writing main code
      i. You'll be writing more code for the purposes of testing your original code than the original code itself
   j. "*Code a little, Test a little*"

3. Basics of Software Engineering

a. Software is becoming so prevalent in nearly everything we do
b. Failures impact everyone
c. Cost of failure becoming *very high*
    i. Financial
    ii. Loss of life
    iii. Time
    iv. Loss of equipment
    v. Inconvenience
d. Describe what Software Engineering is
    i. "Technological"
    ii. "Managerial"
    iii. "Systematic"
    iv. Development that is "on time" and "within cost estimates"
e. Programming in the large vs Programming in the small
f. Software Development Lifecycle and Phases
    i. Requirements
    ii. Design
    iii. Integration
    iv. Coding/Implementation
    v. Maintenance
    vi. (… don't forget testing *throughout!*)
g. Relative cost or time per phase
    i. Which phase takes up the biggest chunk? Be able to explain why
h. Requirements
    i. Client point-of-view
    ii. Objective
    iii. Validation tests
i. Functional requirements
    i. *"What"* the system or application does (describes a function or behavior)
j. Non-functional requirements
    i. *"How"* the system does things (qualifies a given functional requirement)
    ii. A *"quality"* or property the product has (e.g. efficiency)
k. Constraint – a *design requirement* directly from the *client*
l. Cost increases as faults are found later / importance of front-end

Some Suggested Resources/References:

1. [Module 1] Data conceptualization and Python
   - 01-Data Conceptualization.pdf
   - 01-The Challenger-An Information Disaster.pdf
   - Python 3-X Basics.pdf
   - Module exercises
   - Python Scripts 1-13
   - Homework 1: Data Quiz
2. [Module 2] Python Part II
   - 02-Python - Map Filter Reduce.pdf
   - Python Scripts 14-21
   - Python Scripts sphere, spheretest, newton, newtontest, newtontest2
   - Homework 2: Python
3. [Module 3] Python Part III
   - 1-Private, Protected, and Public in Python.pdf
   - 2-Delete in Python.pdf
   - 3-Functions and Parameter List in Python.pdf
   - Python Scripts 22-26
   - Python exception hierarchy: https://docs.python.org/3/library/exceptions.html#exception-hierarchy
4. [Module 4] Intro. To Searching, Sorting, and Data Wrangling
   - 03-Additional Resource - Binary Search.pdf
   - pyScript27_SortingIntro.py
   - Review the resources for NumPy and Panda (4.6)
   - 04-Brief Introduction to NumPy.pdf
   - 05-Brief Introduction to Pandas.pdf
   - Querying data Discussion exercise:
     - Key_Measures_December_2013_Mod.csv
     - read_dataMod.py
5. [Module 5] Software Testing and Debugging
   - AStudent_Class_courses_test.py
   - AStudent_Class_numCourses_test.py
   - AStudent_Class.py
   - primes.py
   - primes_fail1.py
   - primes_fail2.py
   - primes_test.py
   - Behind Pythons unittest-main.pdf
6. [Module 6] Software Engineering
   - 08-SWEngineering-(1).pdf
   - 08-Functional and non-functional reqs.pdf
   - 08-Functional and non-functional reqs - Solutions.pdf

Notes:

- It might be clear but worth stating, knowledge on *how to use or install* Anaconda or Spyder ***will not*** be tested on this or any future exam
- Advanced printing techniques (including "format string") ***will not*** be explicitly tested on the exam, however you're welcome to use it as part of your solution if you wish
- Redefining/redirecting standard output (**stdout**) to print from the screen (standard output) to a file (see `pyScript06.py` and other related scripts) ***will not*** be explicitly tested on the exam, however you're welcome to use it as part of your solution if you wish
- Web Scraping / Web Crawling ***will not*** be explicitly tested on the exam.
- Specifics of NumPy / Pandas ***will not*** be explicitly tested on the exam – only high-level questions may be asked.

--*--

Like the first exam, Exam 2 will take place online on the Ford testing site. Remember, clicking on the exam link will open up the exam immediately and the timer will begin (so only click the provided link once you're ready to begin!) Exam 2 is also open-book/open-notes but absolutely no collaboration is permitted – it must be individual work. Always state assumptions next to the question you are answering when in doubt.

Good Luck! ☺

--*--