Advanced Python Programming
Spring 2022

Final Exam Guide

Test on 04/27/2022 12:30 PM EDT, on paper, in person, in class

The test consists of

1. 10 multiple choice questions - 30 points
2. 1 Code writing questions - 25 points
3. 5 Short Answer questions - 40 points
4. 6 Single Line Response questions - 25 points

General details:

- The exam can be solved in about 60 minutes.
- The exam is supposed to be individual work. You will be asked to sign an Honor Code statement.
- You will also receive a Canvas announcement about the exam with these details on Tuesday, 04/26/2022.
- The Exam is closed-book, closed-notes. You are **NOT** allowed to use any electronic devices, including graphing calculators. We will have 45 4-function calculators on hand and you may borrow those if required.
- You will have an opportunity to earn 20 extra credit points.
- The exam will examine your understanding of the material rather than your capacity to memorize copious amounts of information.
- Please try and attempt all questions. You get points for trying.
- For Problem 3, we need quick explanations on concepts. The problem could also reading code and finding the output.
- For Problem 4, we only need a single line answer. Please do not waste time explaining. This problem includes quick definitions, explanations, etc.
- For Problem 2, you will be graded based on
  - Whether your code works
  - Are you using the required concepts?
  - Are you adhering to standard programming practices?
  - IT IS MORE THAN JUST AN OUTPUT MATCHING PROBLEM
- Solutions to the short answer questions must be SHORT and to the point. (about 50 words/ 4-5 sentences MAXIMUM). A 6 paragraph essay when 4 lines would do means your answer will just be skimmed through in the interest of grading time (The needs of the many ...)

- Anything from the homeworks / quizzes / in class examples / exercises / slides is fair game. You don't need to look for more material.
- Making me laugh might gain you points (depends on the quality of the joke).
- **General Advice: For the sake of completing the assignment on time, please answer all questions before seeking to verify it using course material. Attempting this like a closed-notes test would be best for all parties concerned**.

Topics to study

- Python Fundamentals
  - Basic Python Syntax
  - Numeric and Sequence Data types
  - I/O - print statements and the input() function
  - Selection statements (if - else) and loops (while and for loops).
- Modules, functions and lists
  - Python functions, the "def" keyword
  - Different ways of passing arguments - positional arguments, keyword arguments and packing
  - Modules and importing modules
  - List Comprehension
  - Lambda Functions - Filters, Map/Reduce
- Python Built-In Data Structures
  - Lists - Building other data structures using lists
  - Tuples
  - Sets
  - Dictionaries
- File input and output
- Exceptions - raise and except
- Python Strings
- Object Oriented Programming with Python
  - Closures and Decorators
  - Classes - Defining classes, data attributes and instance methods, static attributes and methods, constructors, inheritance
  - Iterables and Iterators
  - Generators
- The Python Standard Library
  - Built-in Constants
  - time
  - system

  - os
  - re
  - copy
  - itertools
- Python Devtools
  - virtual environments
  - documenting
  - logging
  - Unit Testing
- Full-stack Development
  - JavaScript Frameworks for the Front-End
  - Django - Python support on the Back-End
  - Databases for storage and persistence
  - NOSQL databases
  - MongoDB
- Machine Learning and clustering
  - Supervised and Unsupervised learning
  - Hierarchical and K-means clustering
  - pandas and sklearn
  - Introduction to Pattern Recognition and Neural Networks
  - Working with torch for Neural Networks
  - Building random Sample datasets
  - Building Datasets through Web Scraping
- Building GUI Applications
  - Qt and Widgets
  - Even Driven Programming - Signals and Slots
- Concepts of SaaS Struts
  - Introduction to Virtualization
  - Containers and Docker
  - Computing on the Cloud - AWS EC2 and AWS Lambda
- Python Optimization
- The multiple choice and the output questions will test your familiarity with the Python language and syntax. The code writing questions will test you knowledge of programming.
- Studying the topics listed above will be enough to pass the test. To get a 100, you would be required to study everything on the notes.
- You don't need to study from outside sources. The test is made entirely from the notes and assignments.

Some Sample Problems

1. What is the output of the following code? Explain why.

```
def to_upper(k):
    return k.upper()
x = ['ab', 'cd']
print(list(map(to_upper, x)))
```

2. What is the length of `sys.argv`?
   (a) number of arguments
   (b) number of arguments + 1
   (c) number of arguments – 1
   (d) none of the above
3. List 3 advantages of virtualization.
4. List 2 ways to connect to an AWS instance
5. Explain the theoretical concept behind Supervised Learning
6. In the GUI context, what is a Widget? Can they by layered?
7. Given the root of a binary tree, return the most frequent subtree sum. If there is a tie, return all the values with the highest frequency in any order.

   The subtree sum of a node is defined as the sum of all the node values formed by the subtree rooted at that node (including the node itself). The number of nodes in the tree is in the range $[1, 10^4]$.

   Sample Run 1

```
Input: root = [5,2,-3]
Output: [2,-3,4]
```

   Sample Run 2

```
Input: root = [5,2,-5]
Output: [2]
```