

The Final will be Wednesday, May 6, from 1 - 2:50 PM in our normal classroom. It will be open book and notes. The test will be comprehensive in nature with respect to the programming aspects of the course. Review topics for Test 1 are at:

<http://www.cs.uni.edu/~fienup/cs051s09/lectures/review1.pdf>

New review topic on the Final since Test 1 are:

Chapter 6.

- Random numbers functions from the `random` module: `randint`, `randrange`, `random`, `uniform`
- Value-returning functions: `syntax`, `return` statement, returning multiple values
- Numeric functions from the `math` module (I'll give you a table like 6-2)
- Storing your own functions in a module
- Menu-driven programs

Chapter 7.

- File concepts: input and output files, three general steps (open, process, and close file), types of files (text and binary), file access methods (sequential and direct access), filenames and extensions, and file objects
- Python file `open` function and some modes (`'r'` , `'w'` , `'a'`)
- Python file `close` function
- File object methods: `write`, `read`, `readline`
- concatenating a newline when writing a line to a file (`... + '\n'`)
- stripping a newline from a line of read from a file using the `.rstrip('\n')` string method
- reading and writing numeric data from/to a text file.
- using loops to process a file: `for` loop and detecting the end-of-file condition in Python
- processing a file of records
- handling exceptions using the `try/catch` statement

Chapter 8.

- concept of a sequence in Python: strings, lists, and tuples
- general operations on sequence types: `len`, `min`, `max` functions, indexing, slicing, testing for a sequence item using `in` and `not in`, concatenation using the `+` operator, the repetition operator `*`, iterating over a sequence using a `for` loop, and the `IndexError` exception
- strings are immutable
- string methods for testing strings: `isalnum`, `isalpha`, `isdigit`, `islower`, `isspace`, `isupper`
- string methods that return modified versions of themselves: `lower`, `upper`, `rstrip`, `lstrip`, `strip`
- string methods for searching and replacing: `endswith`, `find`, `replace`, `startswith`
- lists are mutable so you can assign a new value to an individual index
- list methods: `append`, `index`, `insert`, `pop`, `del`, `count`, `remove`, `sort`, `reverse`
- copying a list vs. assigning a list, passing a list as a parameter, returning a list as a parameter
- processing a list of numeric values: summing all items, averaging all items, finding max or min
- processing a list: sequential and binary search

- simple sorting algorithms on a list: selection sort, insertion sort, bubble sort
- working with lists and files: string `split` method, some basic `re` module concepts and methods

Chapter 9.

- concepts and terminology of object-oriented programming (OOP): objects, classes, data attributes, methods, encapsulation, data hiding, UML diagrams
- OOP in Python: class definition syntax, initializer method (“`__init__`”), class methods (e.g., accessors and mutators), operator overloading (e.g., `__add__`, `__str__`, etc.), storing classes in modules
- working with multiple instances/objects: list of objects

Chapter 10.

- concepts and terminology of inheritance: generalization and specialization, “is a” relationship, subclasses, superclass, polymorphism (i.e., subclass method overrides the superclass method)
- Inheritance in Python: subclass definition syntax, calling the superclass initializer method (“`__init__`”), polymorphism of a subclass method to override a superclass method of the same name, `instanceof` function

Chapter 11.

- concepts and terminology of recursion: base case(s), recursive case, direct vs. indirect recursion, divide-and-conquer algorithms, run-time stack, depth of recursion, call-frame, dynamic programming
- recursion vs. looping: efficiency considerations
- Examples: binary search, fibonacci, merge sort, quick sort

Chapter 12.

- concepts and terminology of Graphical User Interfaces (GUIs): command-line vs. event-driven GUIs, `mainloop`, callback function/event handler
- Using the Tkinter Module widgets: `Frame`, `Toplevel`, `Label`, `Button`, `Entry`, `Listbox`, `Radiobutton`, `Checkbutton`, `Menu`, `Menubutton`, `Canvas`, `Scrollbar`, `Text` and their options (`text=`, `command=`, `textvariable=`, `label=`, `variable=`, `height=`, `width=`, etc.)
- Layout management methods: `.pack`, `.grid`
- Control variables: `StringVar`, `IntVar`, `DoubleVar` and their methods `.get()` and `.set(newValue)`