Exam Topics:
* classes and objects, inheritance, object-oriented analysis
* ADTs: stacks and queues
* Exceptions
* Testing; unittest
* Linked lists
* Recursion
* Trees: general, binary, BSTs
* Memory model
* Priority queues and heaps
* Sorting algorithms; running times
* Anything from labs, exercises, project

Running Times
* goal: abstract measure of "running time"
* count "number of operations" executed by the algorithm/program
* this is different for every input: it's a function of the input
* convention: express this as a function of the input's "size"
  (because every type of input has a size)
* convention: use worst-case-- over all inputs of the same size
  (easy to compute; gives a guarantee)
* want to know: rate of growth of worst-case running time
* use big-Oh/big-Omega/big-Theta notation to express running time
to within a constant factor

Examples:

```python
def nonsense1(L):
    a = 0
    for i in range(len(L)):
        if L[i] < a:
            a = L[i]
    return a
```

\[ n = \text{len}(L) \]

\[ O(n) \text{ operations} \]

\[ O(n) \text{ iterations } (i=0,1, \ldots, n-1) \]

\[ O(1) \text{ ops.}, \text{ meaning some constant number} \]
def nonsense2(L):
    a = 0
    for i in range(len(L)):
        b = 0
        for j in range(i):
            b += L[j]
        if b < a:
            a = b
    return a

\[
O(n^2) \left[ \begin{array}{l}
\text{for } i \text{ in range(len(L))}: n \text{ iterations}
\text{for } j \text{ in range(i)}: \text{ i iterations...}
\text{if } b < a: \text{ O(1) bound }
\end{array} \right] O(n) - \text{upper bound}
\]

\[
\# \text{ops} \approx k \cdot 1 + k \cdot 2 + \ldots + k \cdot (n-1)
= k \left( 1 + 2 + \ldots + (n-1) \right)
= k \frac{n(n-1)}{2} \leq k' n^2
\]

def nonsense3(L):
    a = 1
    i = 1
    while i < len(L):
        a *= L[i]
        i *= 2
    return a

\[
\mathcal{O}(\log n) \left\{ \begin{array}{l}
\text{while } i < \text{len(L)}: \text{ # iterations?}
\text{a *= L[i]}
\text{i *= 2}
\end{array} \right\] O(1)
\]

\[
\log_2 n = \frac{\log_{10} n}{\log_{10} 2} \text{ constant!}
\]

\[
k \gg \log_2 n
\]