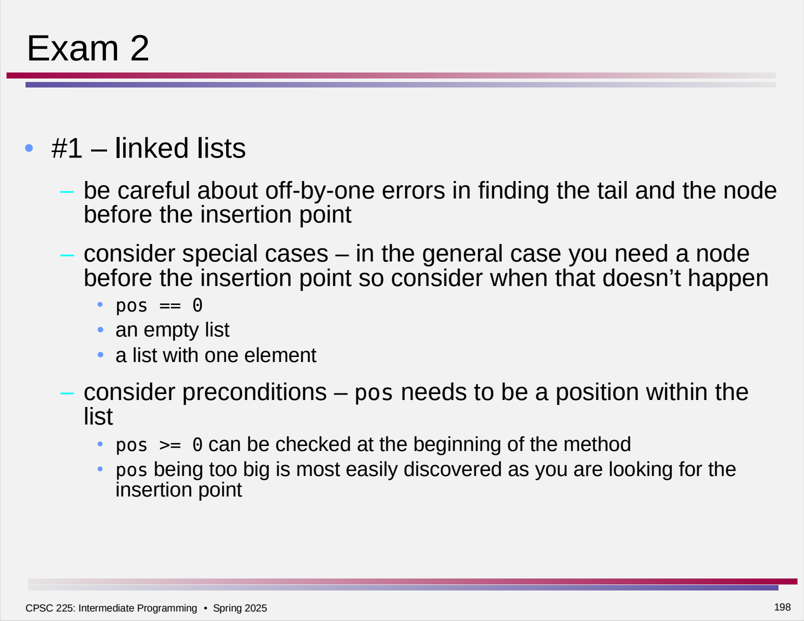
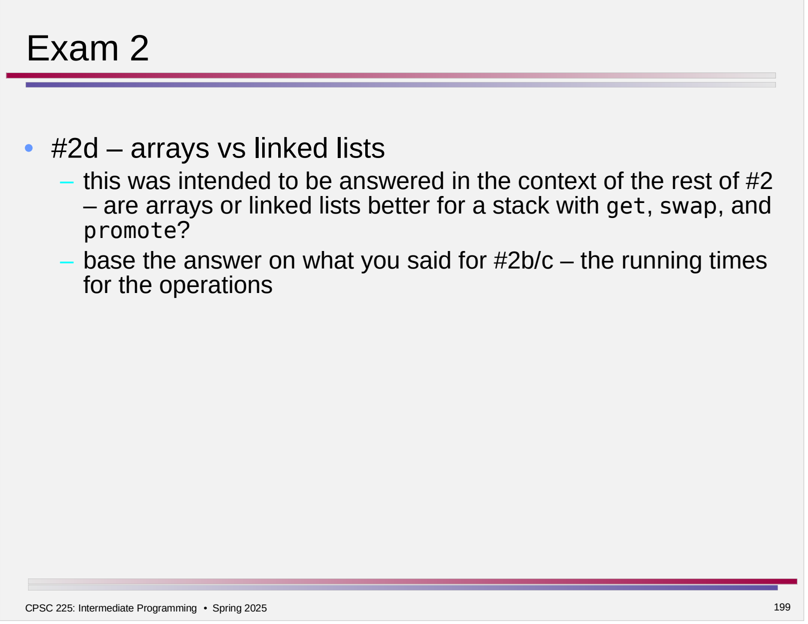


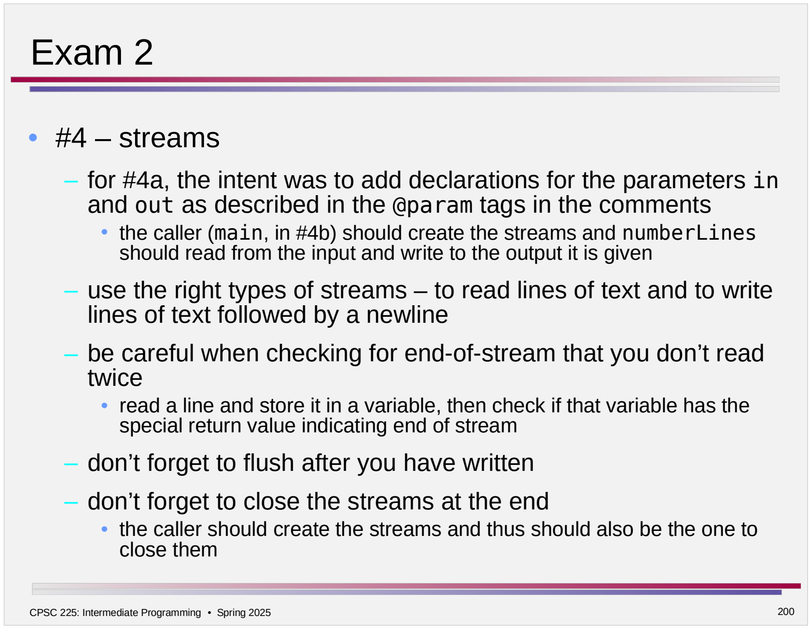
Exam 2

- #1 – linked lists
  - be careful about off-by-one errors in finding the tail and the node before the insertion point
  - consider special cases – in the general case you need a node before the insertion point so consider when that doesn't happen
    - `pos == 0`
    - an empty list
    - a list with one element
  - consider preconditions – `pos` needs to be a position within the list
    - `pos >= 0` can be checked at the beginning of the method
    - `pos` being too big is most easily discovered as you are looking for the insertion point



Exam 2

- #2d – arrays vs linked lists
  - this was intended to be answered in the context of the rest of #2 – are arrays or linked lists better for a stack with `get`, `swap`, and `promote`?
  - base the answer on what you said for #2b/c – the running times for the operations



Exam 2

- #4 – streams
  - for #4a, the intent was to add declarations for the parameters `in` and `out` as described in the `@param` tags in the comments
    - the caller (`main`, in #4b) should create the streams and `numberLines` should read from the input and write to the output it is given
  - use the right types of streams – to read lines of text and to write lines of text followed by a newline
  - be careful when checking for end-of-stream that you don't read twice
    - read a line and store it in a variable, then check if that variable has the special return value indicating end of stream
  - don't forget to flush after you have written
  - don't forget to close the streams at the end
    - the caller should create the streams and thus should also be the one to close them