CS 116 Midterm Post Mortem:

Overall the exam was fairly well done but here were some of the more common mistakes:

Q1:

Q1b:
Overlooking the $x > 30$ clause

Q1c:
Recall that slicing does not include the endpoint so we actually get "nea" instead of "neat". Many students gave the empty list or None instead of the empty string.

Q1d:
Missing that $x,y$ are aliases.

Q1e:
Missing mutation of $x$.

Q1f:
Missing mutation of $x$.
Not assigning None to $y$.

Q2a:
Note that not having a return statement is not an error! Adding None to an integer is however.

Q2b:
Range can consume 1 integer.
Lists can have elements of differing types
Some students were too vague in their answer. Really needed to point out that length 10 lists are indexed from 0 to 9 and so the index 10 is out of range.

Q2c:
Many students who got this wrong failed to mention strings are immutable.

Q2d:
Some students wrote that sqrt does not consume an integer (it can) and some people mentioned the answer cannot be fully computed (which is not an error).

Q2e:
Some student wrote missing an else statement is an error but this is incorrect. Indentation is the actual problem.

Q3a:
Using find instead of count
Reversing which is first

Q3b:
Off by one errors.

Q3c:
Reversing with .reverse(), which mutates and returns None.
Adding 1 instead of subtracting 1.

Q3d:
Using map instead of filter.
Messing up lambda syntax (omitting lambda entirely).

Q4:
Note list slicing returns a list so for example, L[::2] is ['abcd', ['kl', 'mn']]
Many off by one errors.
Many students forgot quotes around strings or list brackets around lists.

Q5b:
A number of students got 33 instead of foo(33) which is 31.

Q5c:
Many students only wrote one line printed or had the lines in the incorrect order. Remember foo(30) doesn't print until foo(35) is resolved and foo(35) => 33 prints first. Then foo(33) must get resolved before foo(30) is returned and so foo(33) => 31 also prints before foo(30) => 31 prints.
Many students also believe that printing includes strings which is not the case. Printing removes the quotes used to denote strings.

Q6a:
Typically students not earning full marks missed one of mentioning the input parameter, return description, printing or consuming input.

Q6b:
Missing print or input were the only mistakes.

Q6c:
Putting too many parameters in the type.
Having an incorrect codomain of None instead of Float.
Placing requirements on the input. Requirements should only be for parameters in the contract.

Q6d:
Not using input at all to read in a value.
Not remembering that the value returned by input is a string, so you have to check for string values, or you need to convert the value to an integer.
Forgetting to return or print, or putting the return before the print statement.
Small issues with the print statement (not converting an integer to a string when using + to create the string to print, for example).

Q6e:
Not setting the correct number of arguments to set_input.
Setting too many arguments to set_print_exact (recall the prompt is not tested by our module!)
Not using check.within.
Not using floating point values or the given tolerance.

Q7:
Incorrect base case (returning False instead of True or off by one errors such as len(s) == 1 instead of if len(s) == 0 or s == '').
Accessing elements out of index ranges (usually due to incorrect base case).
Accessing s[i+1] when i == len(s)-1.
Not returning the recursive call.

Q8a:
Summing even integers
Using map instead of filter
Omitting a return statement.

Q8b:
Not using accumulation: often using a counter but no accumulator.
Omitting even/odd check.
Returning the sum of the evens.
Incorrect base case.

Q9:
Trying to mutate the list by recursing over the rest of the list, adjusting the first position on each recursive call. This doesn't mutate the original list beyond position 0.
Creating a new list of updated values, and then saying L = new_list at the end. This does not mutate the original list either.
The solution required a helper function recursing over positions in the list, and making assignment statements to the list positions.

Q10:
Many students thought .strip() would remove all spaces but note that this only removes spaces from the front and back of a string.
Off by one errors were also common (using 3 or 5 instead of 4 in slicing).