Here is a list of common errors provided by the graders for CS135: Midterm 1.

Question 1

Q1c
- Missing `define`.
- Extra brackets. Remember defining a constant is not the same as defining a function. For instance, `(define (tax-rate...))` has too many brackets.

Q1d
- Capitalize types.
- Missing function header.

Q1g
- Remember that `and` produces `false` if any earlier parameters get evaluated to `false`, so no error occurs.

Question 2

Q2a
- Remember to use the constants `pi` and `e`, as they are predefined.
- Make sure order of operations are correct.
- Ensure that you have enough brackets to close the function definition.

Q2b
- Be sure to know the difference between `sqr` and `sqrt`. `sqr` means squared, and `sqrt` is square-root. Also know the order to apply them. Since square is on the outside, it needs its parameter to be a value before evaluating it, so that is why we do square root on the inside first before squaring everything.

Question 3
- The `[final]` is the final answer, not the 3rd step.
Q3a
- It is incorrect to substitute multiple instances of a constant in an expression. We only do this for a function call when all its arguments are values. For expressions, do the substitutions from left to right one at a time.
- When evaluating expressions, do them from left to right and you move on to the next one once they are a value (cannot be further simplified).

Q3b
- Substituting function arguments in the body occur in one step. And this only happens when all parameters of the function call are values.

Q3c
- If a function’s arguments are constants, you must substitute them in one step at a time.

Question 4
- [final] means final answer, not the 3rd step.

Q4a
- When tracing and and you have a true argument, remember the next step is to remove it, not to move to the next expression.

Q4b
- boolean? evaluates to true if its argument is true OR false, and false otherwise.

Q4c
- or does not have to have a true argument. Similar to and if or has a false argument, remove it before proceeding to the next step.

Question 5
- When evaluating question-answer pairs, note that the previous question(s) (if any) must have evaluated to false.

Q5a
- Using inequalities with 3 arguments was not the desired solution.
- If you checked (and (>= a b) (>= a c)) then the next question (to check if b is the max) can simply be (>= b c), because you already know a cannot be the maximum.
Q5b

- (and a? true) is unnecessary. Simply just use a? which is a boolean value itself you can use for question-answer pairs.

- If you checked (and b? a?) then you can just do a? for the next question instead of checking for (and (not b?) a?).

Question 6

Q6a

- Many students were missing the function name, colon, and/or arrow.

- Many students used Num instead of Nat for parameters a and d. Or, used Int without stating the range, which is not less than 0.

Q6bcdef

- Many students had an off-by-one indexing error when using substring.

Question 7

Q7a

- General punctuation errors, such as missing semicolons (;) at the beginning or missing colons (;) after the function name.

- Incorrect types listed (for instance, Any Any -> Any).

- Incorrect function names.

- Parentheses being added where there should be no parentheses.

Q7b

- Some students were unfamiliar with the string functions used in this question (string-length, string<?, string>??, string<=?, and string>=?). Note that the length function itself, as well as mathematical comparison operators (<, >, <=, >=) cannot be used on strings.

- There were also some instances where logic was reversed (i.e., the first string was produced when the second string should be produced, and vice versa). Take careful note of what you produce when comparing.

- Occasional usage of else-cond, which is bad style.

Q7c

- Many students listed 5 cond questions, largely due to needing an additional check for equal string lengths. However, it is possible to further reduce the number of conds by reordering the checks conducted.
Q7d

- Many students had only one test case that checked for different length strings and same length strings by assuming that one of the input arguments was always going to be alphabetically first (instead of also including the cases where the second argument could also be alphabetically first).

Q7e

- Many students included redundant tests (i.e., multiple tests that effectively checked for the same thing).
- Many students assumed they only need one test for a Bool output instead of providing two separate tests (one that produces true, and one producing false).

Q7f

- If both arguments were Num, some students incorrectly produced the maximum of the two arguments instead of the minimum.
- If both arguments were Bool, many students lost marks by overcomplicating the code for this conditional.
- If both arguments were Str, many students lost marks by forgetting they can use the previously implemented function string-min or forgetting to pass arguments to the string-min function.