For #1–5, "show your work" applies to (a) and (d).

- For (a), there's an algorithm for finding candidate keys. Show evidence of following that process and of computing closures even if the problem is small enough that you can figure out the keys in your head.
- For (d), if more than one FD violates the normal form, show the decomposition after each FD rather than just the one final result.

For (c), list all of the FDs that violate the normal form.

For (d), make sure you answer is complete:

- For each decomposition, be sure to check that the result is lossless and dependency preserving.
- Don't forget to address why the end result is fully normalized.

Show PKs in all of the schemas (along the way of the decomposition and in the final result). It is OK to only list FKs for the final schema to avoid redundant writing.

Choosing between candidate keys: any candidate key *can* be used as PK — they all serve to identify rows, however, shorter keys (fewer attributes) is preferred — every FK has to refer to the whole PK. For #2b, the student ID is likely to be a more stable value (email addresses can change), so it is preferred. For #5b, enrollment is about the student and the course so `(studentID,course)` is more natural — that `(studentID,instructor)` is also a candidate key is a consequence of an instructor only teaching one course and so it it is more fragile.

List all the FDs that violate a normal form — in #5 some decompositions are dependency preserving while others are not. You may need to try several different decompositions to make as much progress as possible.