Answer

In some ways, this question goes a bit beyond the lecture. It discusses what can go wrong, particularly for a non-recommended deployment. That said, this list of answers includes things that people do try to implement from time to time, so we want to make sure you're aware of the dangers.

Answers:

* Reading from a secondary prevents it from being promoted to primary.
  + False.
  + Reading from a secondary does not directly affect a secondary's ability to become primary, though if the reads caused it to lag on writes and fall behind on the oplog, that might make it ineligible until it is able to catch up. [Here's a note on replication lag](https://docs.mongodb.org/manual/tutorial/troubleshoot-replica-sets/#check-the-replication-lag).
* If the secondary hardware has insufficient memory to keep the read working set in memory, directing reads to it will likely slow it down.
  + True.
  + This could really go either way. If the secondary has excess capacity, beyond what it needs to take writes, then directing reads to it would cause it to work more, but perhaps it would still be able to keep up with the oplog. On the other hand, if the primary is taking writes faster than the secondary can keep up, then this scenario would definitely slow it down.
  + Generally, your secondary should be on the same hardware as your primary, so if that's the case, and your primary *would* be able to keep up with the reads, then this shouldn't be a problem. Of course, if your primary can handle both the read and write loads, then there's really no compelling reason to send the reads to the secondary.
* If your write traffic is great enough, and your secondary is less powerful than the primary, you may overwhelm the secondary, which must process all the writes as well as the reads. Replication lag can result.
  + True.
  + This is a design anti-pattern that we sometimes see.
  + A similar anti-pattern occurs when reads are routed to the primary, but the secondary is underpowered and unable to handle the full read + write load. In this case, if the secondary becomes primary, it will be unable to fulfill its job.
* You may not read what you previously wrote to MongoDB on a secondary because it will lag behind by some amount.
  + True.
  + This is pretty straightforward. Unless you are reading from the primary, the secondary will not necessarily have the most current version of the documents you need to read.
  + Whether this is a problem or not depends on your application's requirements and business concerns, so it goes a bit outside the scope of development.