1. List and describe at a high level the steps involved in the requirements engineering process.  
   (pg 104)

* Elicitation – gathering requirements using (more than one of) techniques like interviewing, observing, brainstorming, role playing, etc. to uncover unspoken rationale/goals
* Documentation and definition – formalizing requirements in writing using diagrams and unambiguous terms
* Specification – a formal document that provides introduction, high-level description, and detailed descriptions of I/O, interface, performance characteristics, constraints/standards, “-ity” traits, etc.
* Prototyping – usually concerned with look and flow of the user interface
* Analysis – categorizing and prioritizing requirements; looking for consistency and completeness
* Review and validation – to reduce design/programming errors and catch large issues early, while not straining resources that can go towards other phases of the overall project.
* Agreement and acceptance – formal exit from requirement to design phase

1. What are the three main items that must be planned prior to conducting requirements engineering?  
   (pg 103)

* For resources
* For methodology
* For time needed

1. What are the six main dimensions of requirements that you need to address when collecting requirements? Briefly describe each dimension. (pg 110 – 112)

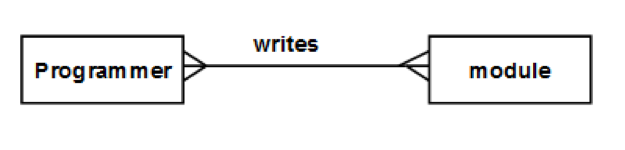
* Individual functionality – discrete activities that the system must perform
* Business flow – the context in which the discrete activities above take place
* Data, formats, and information needs – what inputs and outputs are produced by an application, how each of those should be structured, and what those data mean to various users
* User interfaces – visual layout and controls that facilitate business flow
* Systems with other interfaces – protocols/standards for transferring control or data to external applications/systems
* Other constraints such as performance, reliability, and security – “nonfunctional” requirements that are not about “what” an application/system does, but the attributes it must have or metrics it will be subject to

1. List the items that are included in the description of a high-level business profile. Briefly describe each item. (pg 108-110)

* Opportunity/needs – a business-oriented problem that may or may not include software, but usually involves a cost
* Justification – the benefit/return of investing in a project to solve the problem
* Scope – the area of focus for the current solution, limiting what the project will deliver
* Major constraint – things like budget or schedule that dictate the size/pace of a solution project
* Major functionality – most important business activities that the software will perform
* Success factor – staging releases with awareness of business priorities for maximum impact
* User characteristics – knowing the responsibilities and capabilities of the day-to-day users of the software

1. List and describe three items that you will need to consider when prioritizing requirements.  
   (pg 117)

* Current customer needs – A product or feature is more likely to succeed in the market and generate revenue if we know for sure that there is demand for it.
* Competition and current market condition – Only if we can do something better, or cheaper (at the same quality) as an existing process/solution, then we have a chance of users/customers adopting what we produce over what others produce.
* Future customer needs – A product or feature may be focused on solving or improving only one important area of a series or network of “lesser” concerns which could all eventually be addressed.

1. Explain in an ER diagram the relationship between programmers and modules where a programmer may write several modules and each module may also be written by several programmers.   
   
2. What are the four types of requirements traceability? Briefly describe each one. (pg 120)

* Backward from traceability: from the requirement to the document source or creator
* Forward from traceability: from the requirement to design and implementation
* Backward *to* traceability: from the design and implementation back to the requirement
* Forward *to* traceability: from the document/creator to the requirement

1. Technical Analysis and Summary (requires additional effort). Compare and contrast traditional requirements engineering as presented in our textbook with how requirements are handled in Scrum based on your experience and what was presented in class. If you use text or images from other sources, proper credit/attribution must be noted.  
   In Scrum, anyone on the product team can add a story (a.k.a. ”requirement” in traditional requirements engineering) to the product backlog at any time, while in traditional requirements engineering there may be specialists like business analysts and test engineers who need to give input before the start of the design phase. In Scrum, there are no entry criteria for the requirement phase, but a user story cannot be pulled into the design/development phase during sprint planning unless it is complete with an estimate of story points, acceptance criteria, a list of technical tasks, and a definition of done. In both Scrum and traditional requirements engineering, prioritization methods are similar but Scrum allows more frequent opportunities to reprioritize. In Scrum, it is up to the development team to schedule requirements engineering activities as necessary whereas traditional requirements engineering needs a more formal plan as part of an overall waterfall process.