1. (2 points) The text describes two aspects of “the problem” of building and delivering software systems on time. What are these two aspects? (You don’t need to describe these aspects, just name them.)

   Change and Complexity.
   (See page 3.)

2. (3 points) In defining software engineering, the text describes it as a combination of four activities, one of which is that software engineering is a “rationale-based activity.” List the other three activities?

   Modeling
   Problem solving
   Knowledge Acquisition
   (See pages 5, 6.)

3. (3 points) Briefly discuss the rationale-based activity.

   Rationale is part of the decision making process. Why did you choose design A over design B? At some future time, the reasons for not favoring design A may have evaporated. Maybe there is some third-party software product available. Maybe that database technology has become the de facto standard. On large projects, this type of recording of rationale becomes more important.
   (See pages 9 and 10.)

4. (3 points) What is the difference between a scenario and a use case?

   A use case is a general description of all the possibilities for a particular system functionality; for example, “select product for list.” A scenario is a specific instance of a use case; for example, “Bob selects macaroni and cheese.”
   (See page 46.)

5. (1 point) What does it mean to say that a communication mechanism is synchronous? (Don’t just give an example, give a brief explanation please.)

   The parties involved are participating at the same time (e.g., telephone call).
   (See class notes or page 102.)
6. (5 points) The text says that there are three system models used in software engineering. These are the functional, structural (static), behavioral (dynamic) models. We have learned about five different UML modeling diagrams: use case, class, statechart, sequence, and activity. Indicate which system model each diagram is associated with.

   Functional: Use case diagram
   Structural: Class diagram
   Behavioral: Statechart, sequence, and activity diagrams

7. (4 points) Two representations of the dynamic or behavioral system model are statechart diagrams and sequence diagrams, but they describe different perspectives. The statechart describes the behaviors across all functionalities that are relevant for a single object in a system. What is the perspective of the sequence diagram?

   The sequence diagram depicts the behaviors across all classes/objects needed to achieve a single task (i.e., use case).

   (See page 30.)

8. (2 points) Consider our gas pump system from earlier in the term. Is the following a functional or nonfunctional requirement?

   ‘‘The customer can select one of three ’grades’ of fuel.’’

   Functional. Not being able to choose means that you can only get the default grade determined by the pump’s software. Doesn’t sound good.
9. (9 points) Match each of the software engineering activities listed in the left column below with the phase listed in the right column in which the activity is performed. (Each item in left column matches 0 or 1 item in right column. Each item in right column matches 0 or more items in left column.)

<table>
<thead>
<tr>
<th>Activity</th>
<th>Phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identifying use cases</td>
<td>Req.Elic.</td>
</tr>
<tr>
<td>Identifying boundary objects</td>
<td>Analy.</td>
</tr>
<tr>
<td>Identifying actors</td>
<td>Req.Elic.</td>
</tr>
<tr>
<td>Developing likely scenarios</td>
<td>Req.Elic.</td>
</tr>
<tr>
<td>Identifying subsystems</td>
<td>nothing</td>
</tr>
<tr>
<td>Mapping use cases to objects with sequence diagrams</td>
<td>Analy.</td>
</tr>
<tr>
<td>Identifying applicable software architectures</td>
<td>nothing</td>
</tr>
<tr>
<td>Identifying nonfunctional requirements</td>
<td>Req.Elic.</td>
</tr>
<tr>
<td>Identifying entity objects</td>
<td>Analy.</td>
</tr>
</tbody>
</table>

10. (3 points) Requirements specification documents strive to achieve characteristics of being unambiguous, correct, complete, consistent, and verifiable. Which answer best describes what it means for a requirements specification to be *complete*.

(a) All features of interest are described by requirements.
(b) Repeatable tests can be designed to demonstrate fulfillment of requirements.
(c) A requirement cannot be interpreted in two mutually exclusive ways.
(d) The requirements describe the features of the system and environment of interest to the client and the developer, but do not describe other unintended features.
(e) No two requirements of the specification contradict each other.
11. (2 points) True or False. In software engineering it is important to complete each activity in its entirety before proceeding to subsequent activities. For instance, requirements elicitation activities should not be revisited when analysis activities have begun. True or false.

False. We expect to need to revisit earlier activities, as changes occur and new knowledge is acquired and old knowledge is invalidated.

12. (2 points) Computer science is not a physical science (like Chemistry) and it is not a social science (like Psychology). The text suggests it is something different altogether. What kind of science is it?

Page 6 calls it a science of the artificial.

13. (6 points) Describe the relationship between abstracting something and refining something. Also, give an example.

Abstracting something means ignoring unnecessary details. Refining means adding details. As things become refined they become less abstract. It is possible, however, to relook at a detailed thing in an abstract way.

For example, all the GUI components necessary to build an input form can be abstractly considered as a single package or a single class. Refining this view would introduce things like button objects, textfield objects, label objects, event handlers, etc. Even knowing this detail, we can abstract back out to describe them collectively as the GUI input form again.

14. (2 points) Non-functional requirements have been categorized by some software engineering researchers and the categories are referred to as URPS+. What does the U,R,P, or S stand for?

Pages 126,127. Usability, Reliability, Performance, Supportability.

15. (1 point) Give an example of a scheduled communication event.

A code walkthrough, or a status meeting.
16. (5 points) What are the “parts” of a use case specification? (Hint, the first “part” is the name of the use case.)

Use case name
Participating actors
Entry conditions
Flow of Events
Exit conditions
Quality requirements

17. (5 points) Brainstorm a scenario for using your AppCard at the 3rd floor soda machine.

Student with card approaches machine
Swipes card
System looks up student’s AppCard account, verifies sufficient funds
System displays prompt to make selection
Student makes selection
System debits student account
System distributes selection

18. (2 points) For each of the following subproblems, the two classes are related to each other. Select the best representation of this relationship.

(a) Vehicle
Car
Car inherits from Vehicle, so open triangle line to Vehicle from Car.

(b) Backpack
Book
Books aggregate in backpacks, so open diamond on backpack side to books with a * multiplicity at books.
19. (8 points) In the past, I’ve asked students to suggest how to model years, months, and days. Please “review” (that is tell me if it is okay, or what you think is incorrect/imprecise about it) the following student responses.

(a) 

Probably should be composition rather than aggregation, and multiplicities are imprecise.

(b) 

Composition still; multiplicities are better but still imprecise since could have 12 months of 31 days each!!

(c) Can you do better? Sure you can, so go ahead!
20. (8 points) Consider the following ATM use case diagram. How would you suggest enhancing the functional model to reflect the new requirement that the system accommodate their Spanish speaking customers. If you think the diagram should change, go ahead and make the changes here. If you think the diagram remains okay, then describe how this situation is added to the model.

Adding a bubble for language choice does not seem to be an Entire scenario and so failing the WAVE test.

Also, how to make sure it is the first activity indicates a problem. The first activity of a larger “functionality” AHA so it’s just an internal part of that larger use case!!

I’d leave the diagram alone and just add an event to the flow of events to get language choice.
21. (12 points) Develop a statechart diagram for our traffic light intersection system's `ChangeLights` control object.

Recall that this is a simple 4-way intersection, with only one lane of traffic in each direction (North, South, East, West). There are no “left turn lanes” or left arrow lights. A sensor is buried into the road for each lane of traffic. A timer is used to change lights in the absence of sensor signals.

Let the initial state be when all lights are red to be followed (after an appropriate safety delay for those cars that always seem to “run” the red light) by North and South becoming green.
22. (12 points) How many times you have walked the halls of this building.....BUT have you ever thought of the building as a set of interconnected objects? No? Then now is the time! Let’s see, I’ll help you get started. There are classrooms and lecture halls and research labs and all these are just specific versions of rooms. There are floors and doors and well you can probably get started now...