NAME: ______________________________________________
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

2. (4 points) In defining software engineering, the text describes it as a combination of four activities. What are these activities? (Just name them.)

   Modeling, Problem solving, Knowledge Acquisition, Rationale-driven

3. (6 points) What is the purpose of modeling and when is it useful?

   The purpose of modeling is to abstractly represent a system. This enables us to answer questions, communicate, and reason about the system. Modeling is useful when systems are too large, too small, too complicated, too expensive, or no longer exist.

4. (5 points) How do we typically model exceptional cases in use case diagrams?

   We typically use an "extends" relationship from the special use case to the originating use case.
5. **TicketDistributor** is a machine that distributes tickets for trains. Travelers have the option of selecting a ticket for a single (one-way) trip, in which case the **TicketDistributor** computes the price of the requested ticket based on the area in which the trip will take place and whether the traveler is a child or an adult. The frequent traveler (i.e., commuter) may alternatively select a time card for a week or a month. The train company has a central computer system that periodically updates cost information in the **TicketDistributor** machines. The **TicketDistributor** must be able to handle several exceptions, such as travelers who do not complete the transaction (**TimeOut**), travelers who select the cancel button without completing the transaction (**TransactionAborted**), and resource outages, such as running out of tickets (**DistributorOutOfChange**) or change (**DistributorOutOfPaper**).

(a) (2 points) Is the following requirement functional or nonfunctional: “The **Ticket Distributor** must enable a traveler to buy weekly passes.”
   Functional

(b) (2 points) Is the following requirement functional or nonfunctional: “The **Ticket Distributor** must be written in Java.”
   Non-functional

(c) (2 points) Is the following requirement functional or nonfunctional: “The **Ticket Distributor** must be easy to use.”
   Non-functional

(d) (2 points) Is the following requirement functional or nonfunctional: “The **Ticket Distributor** must provide a phone number to call when it fails.”
   A close call either way. I’d lean to non-functional; if it doesn’t client will likely be only mildly annoyed.

6. (2 points) There are three system models used in software engineering. The object model is one of them. List the other two.
   Functional model and Dynamic (or behavior) model.
7. (8 points) Two representations of the dynamic system model are statechart diagrams and sequence diagrams. Describe the dynamic “perspective” for each of these two types of UML diagrams.

A statechart provides the perspective of a single object/class, seeing all the possible states of that entity and what causes transitions between states.

The sequence diagram provides a perspective of a use case functionality, what classes/objects are involved and how in order to achieve this functionality.

8. (4 points) What is the difference between a scenario and a use case?

A use case is a generalization of a set of scenarios. A scenario is more specific.

9. (1 point) What does it mean to say that a communication mechanism is synchronous?

It means that the participants of the communication must participate at the same time.

10. (1 point) Give an example of a synchronous communication mechanism.

A telephone conversation.
11. (5 points) Match each of the requirements specification terms with its definition.

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unambiguous</td>
<td>A requirement cannot be interpreted in two mutually exclusive ways.</td>
</tr>
<tr>
<td>Correct</td>
<td>The requirements describe the features of the system and environment of interest to the client</td>
</tr>
<tr>
<td>Complete</td>
<td>All features of interest are described by requirements.</td>
</tr>
<tr>
<td>Consistent</td>
<td>No two requirements of the specification contradict each other.</td>
</tr>
<tr>
<td>Verifiability</td>
<td>Repeatable tests can be designed to demonstrate fulfillment of requirements.</td>
</tr>
</tbody>
</table>

12. (4 points) For each of the following subproblems, the two classes are related to each other. Select the best representation of this relationship.

(a) Month — Year

(b) Vehicle — Car

(c) State — County

(d) Backpack — Book
13. (16 points) Recall our last version of the use case diagram of the parking deck gate system.

Your job is to write out the textual use case for a permitted driver to enter using their card. (You can use the following page.)
Use case name: Enter Using Card
Participating Actors: Registered Parker and Parking Registration System
Entry condition: Driver permitted access information is valid, car present sensor is functioning, gate down
Flow of Events:
1. Driver in car approaches gate
2. Driver swipes card
   3. System detects card swipe
   4. System reads identification off card
   5. System verifies identification with parking registration system
   6. System verifies presence of car
   7. System creates a record of this entry that is stored in an internal database to be used when exiting
   8. System raises gate
   9. System detects car moved through gate
  10. System lowers gate
Exit condition: Gate lowered
Quality requirements: Parking Registration system is down, sensor malfunctions, card reader malfunctions (or card can’t be read)
14. (18 points) While I’m making this test up I notice that the World Series of Poker is on television. Hmmm. That gets me thinking. What would an object model’s class diagram for an online game of poker look like? Draw a class diagram, you can use the next page.

Let’s restrict our modeling to the poker variant known as Texas Hold ’Em.

- In this game, each player gets two cards called “hole cards” that only the player can see.
- At this point, a round of betting ensues. Each player can add money to the pot until the betting stops or a player can choose to stop betting and “fold” which means they are out of the current game.
- When betting is completed, the dealer deals three community cards (known as the ”flop”) face up so they can be seen by all the players. Players can use these cards to form their logical "hand" or group of cards.
- After the flop is dealt, another round of betting occurs.
- Then the dealer turns over a single community card face up (known as the ”turn”). Again, players can use this card in their hand trying to make the best logical hand of 5 cards.
- Another round of betting occurs.
- Lastly, the dealer turns over a final community card face up (the ”river” card).
- A last round of betting ensues.
- All players that have not folded when the betting ends then reveal their best hand of 5 cards from the 7 cards available (the 5 community cards and their own 2 hole cards).

Here’s a picture of an online game that illustrates some of this.
15. (16 points) A statechart diagram for a simple 4-way IntersectionController class is given below. We have been asked to change the system to support crosswalks through this dangerous intersection. Poles will be installed with buttons that allow pedestrians to indicate their desire to cross lanes of traffic. Pedestrians will be given highest priority. Safety is our biggest concern, so no lights are allowed to be green while pedestrians are allowed to cross.

Your job is to change this model to reflect the new requirement, you can use the next page if needed.