

No note-sheets, calculators, etc. on this exam. Please read the whole exam before you get started.

1. (30 points) A factory messaging and time-tracking system allows managers to send and read messages while workers can only read messages. Messages can be in more than one mailbox. All employees have exactly one mailbox, and mailboxes are not shared. Messages include a subject and a body. The recipients can see which manager sent the message and when the message was sent. Workers punch in and out as required by contract. Managers do not punch in and out, but they can review time cards showing the number of hours any employee in their division worked and the times they punched in and out. Each division has at least one manager. No worker or manager is associated with two divisions.
 - a. **Circle** the unique nouns and **box** the verbs. Ensure you find them all! It is critical for the next parts.
 - b. **List** all nouns that should not appear as classes in a domain-level diagram.
 - c. **Use** the information from the paragraph to draw a domain-level class diagram for this system. Draw generalizations and associations with multiplicities. Include simple attributes and operations (but without any types) if they are given in the statement.

2. (20 points) **Write a short code example** illustrating temporal cohesion. **Write a single sentence or sentence fragment** explaining why this illustrates temporal cohesion.

4. (40 points) A building has many displays where each display consists of an array of 20 by 100 light bulbs. Individual displays can be programmed to show a message (using the light bulbs), light just one bulb at random (to show it is off but working), or display a random pattern. **Draw** a UML class diagram capturing how the strategy pattern would be used in this case. At a minimum, **include** classes for the building and displays. Be sure to **mark** abstract classes and interfaces as well as to **show** associations (with directions), inheritance relationships, multiplicities, methods, and attributes.

5. (10 points) **Describe** how your design above is open for implementation but closed for modification.