1. **(2 points)** Give two characteristics of a good design.

2. **(2 point)** Cohesion is a qualitative indication of the degree to which a module

3. **(2 points)** The _____ process model is a risk-driven process model.

4. **(2 points)** Describe two advantages of agile process models over waterfall model.

5. **(2 points)** Describe how listeners in Swing are examples of the Observer pattern.

6. **(2 points)** Describe what is regression testing and why is it done?

7. **(2 points)** What is an advantage of top-down integration schemes over bottom-up integration?

8. *(2 points)* Give two circumstances that could arise leading to an *ineffective* inspection process?

9. **(3 points)** In almost all instances, you should "break" your software Design into many modules, hoping to make understanding easier and as a consequence, reduce the cost required to build the software. Describe the trade-off between cost of software and pursuing higher number of modules using a graph.

10. **(3 points)** How does a sequence diagram differ from a state diagram? How are they similar?

11. **(3 points)** What is load testing? Give an example application that might need to be load tested. Be specific about the aspects of the application that need to be load tested.

12. **(3 points)** The Gang of Four design patterns can be classified under three categories. List these categories and give the name of a design pattern in each category.

13. **(3 points)** How does a well structured unit testing strategy improve the maintenance of an application?

14. **(3 points)** Describe the Blob anti-pattern. What is a characteristic indication that the anti-pattern exists in a design? What steps might you take to eliminate a Blob?

15. (Three parts, 12 points total)

Develop an automated airline reservations system (ARS) for Fly-By-Night Airlines. Passengers can make reservations at automated reservation machines (ARMs) that are located throughout cities served by Fly-By-Night. A reservation will include the flight number, departure date and time, reservation type (first class, coach), a seat number, and the price of the ticket. Once reservations are made and credit card information is verified, tickets are printed for the passenger. Passengers can also cancel or change reservations after they are made. All flight data is entered by a designated data entry clerk.

Note: You may make simple assumptions about this problem, based on your own knowledge about airline reservation systems.

a) **(3 points)** Write the detailed textual use case (step-by-step) for a passenger successfully using the ARM to book a reservation.

b) **(5 points)** Draw a UML Design class diagram for the ARS showing the classes, relationships, and multiplicities between classes

c) **(4 points)** Using your class model, create a UML sequence diagram for a passenger successfully booking a reservation.

16. (14 points)

SoftModel Inc. has been contracted to build a system for parking garages attached to the Space Stations of the 22nd century. The first parking garage to be built will be attached to "Space Station Siren" which orbits Titan, Saturn's moon. The parking garage will be able to park up to 100 space vehicles of all types. It will be have 3 air locks which allow the space vehicles to move from airless space into the pressurized atmosphere of the space station. Each air lock can fit either one space ore vehicle or 3 regular (passenger or military) space vehicle. Future parking garages may have more (or fewer) parking spaces and a different number and size air lock.

The parking technology is similar to those currently used on Earth. Drivers can enter the garage with a permit that can be swiped with a card reader. They can also get a ticket by pressing a button at the entrance. The vehicles have automatic arms to get the ticket inside. You don't want to open a window and let the air out. The ticket has a date and time-stamp indicating the date/time of entering the garage. After swiping the permit or getting a ticket, the driver enters an air lock (if there is room) and then enters the garage itself.

The system to be developed by SoftModel should keep track of the number of cars currently in the garage, and in each air lock. It displays signs indicating whether or not the garage is full, and whether or not each air lock is full. Drivers with tickets pay the attendant at a gate before leaving the garage; drivers with permits have a parking fee added to their account when they swipe their permits upon exiting from the garage. Drivers must pass through an air lock to exit the garage. An operator console displays the status of the system including the number and types of (1) parked vehicles, (2) vehicles in air locks, and (3) vehicles waiting for space in the lot or space in the air locks.

The system must keep track of the payments being made, and the amounts due from permit holders. It must also provide summaries to authorized personnel regarding peak garage hours and the number of permits used.

Design a single **state diagram** for the above system. Please state your assumptions clearly (if any).

17. **(10 points)** The following is a partial domain model (requirements class model) that will be used to develop a system to support scoring of judged athletic competitions (e.g., gymnastics, diving and figure skating). There are multiple events and competitors. Each competitor may enter several events and each event must have at least one competitor. Each event has several judges who subjectively rate the performance of competitors in that event. In some cases, a judge may score more than one event. Trials are the focus of the competition. Each trial is an attempt by one competitor to perform his/her best in one event (i.e., a trial is associated with one competitor and a one event). A trial is scored by the panel of judges for that event and a net score is determined. Complete the domain model as stipulated below:

a) **(3 points)** Add multiplicities to the associations (each end MUST have a multiplicity).

b) **(4 points)** The system is required to track the following information: the name and age of a competitor, the date of a trial for a competitor, the difficulty factor of an event and the net score for a trial. Add elements to the domain model that reflect this information.

c) **(4 points)** Each competitor must register for an event that he/she plans to take a trial in. Add an association (including multiplicities) that tracks the events that a competitor registers for.