(3.1) Prepare a class diagram from the object diagram in Figure E3.1.

Figure E3.1 Object diagram for a portion of Europe

(3.2) Prepare a class diagram from the object diagram in Figure E3.2. Explain your multiplicity decisions. Each point has an x coordinate and a y coordinate. What is the smallest number of points required to construct a polygon? Does it make a difference whether or not a point may be shared between polygons? Your answer should address the fact that points are ordered.

Figure E3.2 Object diagram for a polygon that happens to be a square

(3.3) Using your class diagram for Exercise 3.2, prepare an object diagram for two triangles with a common side under the following conditions.
   a. A point belongs to exactly one polygon.
   b. A point belongs to one or more polygons.

Figure E3.4 Object diagram for part of your family tree

(3.4) Prepare a class model to describe undirected graphs. An undirected graph consists of a set of vertices and a set of edges. Edges connect pairs of vertices. Your model should capture only the structure of graphs (i.e., connectivity) and need not be concerned with layout such as location of vertices or lengths of edges. Figure E3.10 shows a typical undirected graph.

Figure E3.10 Sample undirected graph
(6) Figure E3.9 is a class diagram that might be used in developing a system to simplify the scheduling and scoring of judged athletic competitions such as gymnastics, diving, and figure skating. There are multiple events and competitors. Each competitor may enter several events and each event has many competitors.

Each event has several judges who subjectively rate the performance of competitors in that event. A judge rates every competitor for an 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 or her best in one event. A trial is scored by the panel of judges for that event and a net score determined. Add multiplicity to the diagram.

![Diagram](image)

**Figure E3.9 Portion of a class diagram for an athletic-event scoring system**

(3) Add the following attributes to Figure E3.9: address, age, date, difficulty factor, score, and name. In some cases, you may wish to use the same attribute is more than one class.

(7) Prepare a class model to describe directed graphs. A directed graph is similar to an undirected graph, except the edges are oriented. Figure E3.11 shows a typical directed graph.

![Diagram](image)

**Figure E3.11 Sample directed graph**

(6) Prepare a class diagram for the dining philosopher problem. There are 5 philosophers and 5 forks around a circular table. Each philosopher has access to 2 forks, one on either side. Each fork is shared by 2 philosophers. Each fork may be either on the table or in use by one philosopher. A philosopher must have 2 forks to eat.

(7) The tower of Hanoi is a problem frequently used to teach recursive programming techniques. The goal is to move a stack of disks from one of three long pegs to another, using the third peg for maneuvering. Each disk is a different size. Disks may be moved from the top of a stack on a peg to the top of the stack on any other peg, one at a time, provided a disk is never placed on another disk that is smaller than itself. The details of the algorithm for listing the series of required moves depend on the structure of the class diagram used. Prepare class diagrams for each of the following descriptions. Show classes and associations. Do not show attributes or operations:

a. A tower consists of 3 pegs. Each peg has several disks on it, in a certain order.

b. A tower consists of 3 pegs. Disks on the pegs are organized into subsets called stacks. A stack is an ordered set of disks. Every disk is in exactly one stack. A peg may have several stacks on it, in order.

c. A tower consists of 3 pegs. Disks on the pegs are organized into subsets called stacks, as in (b), with several stacks on a peg. However, the structure of a stack is recursive. A stack consists of one disk (the disk that is physically on the bottom of the stack) and zero or one stack, depending on the height of the stack.

d. Similar to (c), except only one stack is associated with a peg. Other stacks on the peg are associated in a linked list.