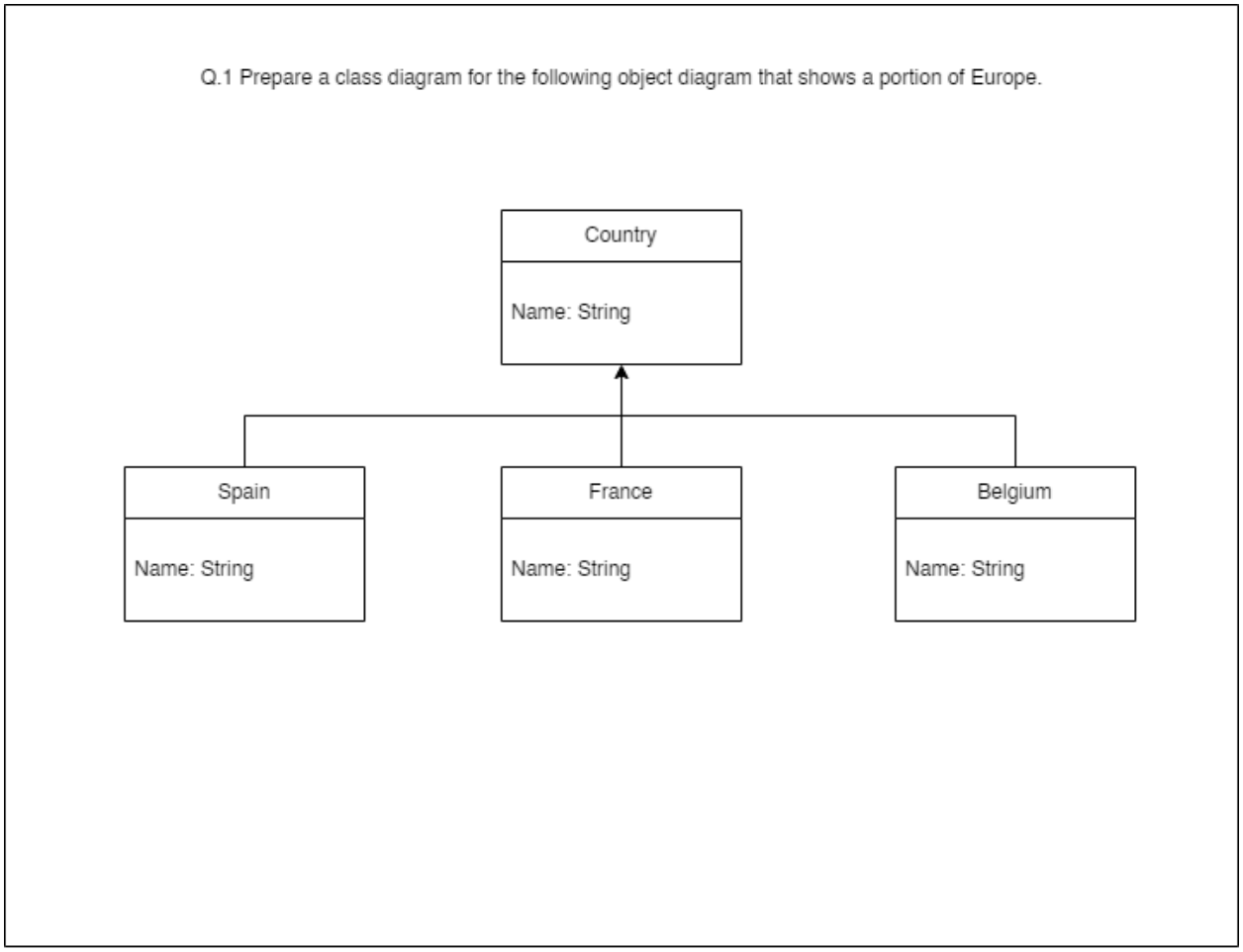
Software engineering

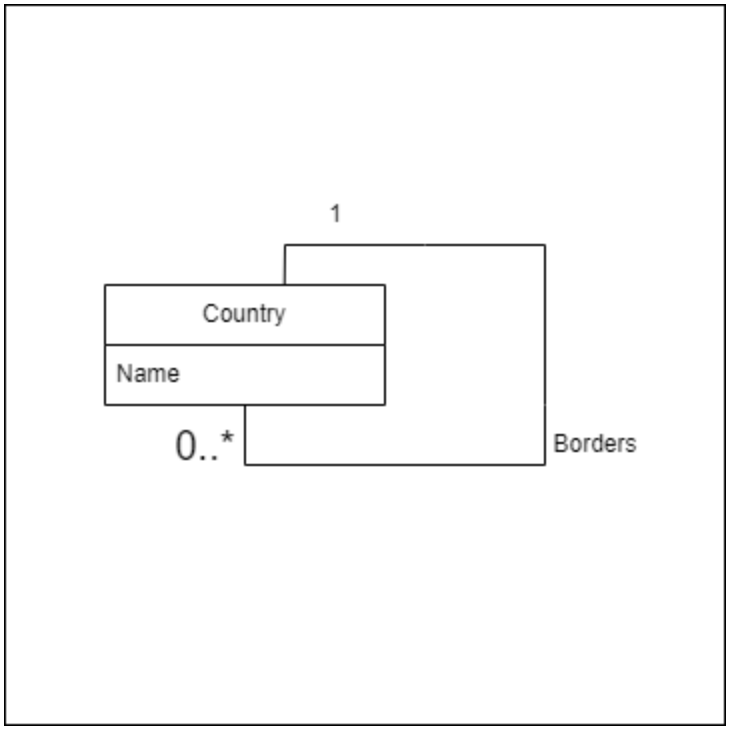
Lab 4 Class diagram

202201151
Pandavadara Sahil Rasikbhai

Q.1 Prepare a class diagram for the following object diagram that shows a portion of Europe.

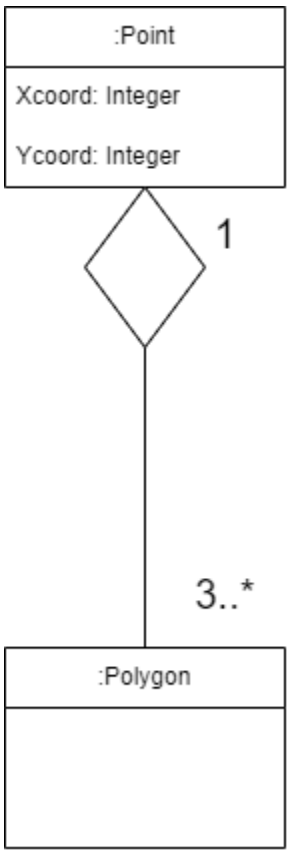
Answer:




1
Country
Name
0..*
Borders

Q.2 Prepare a class diagram for the object diagram given in Figure -2. Explain your multiplicity of decisions. Your answer should address the fact that points are ordered.

Answer:



Sub Que1: Explain your multiplicity of decisions.

Multiplicity Decisions:

- A `Polygon` consists of multiple `Points`. The minimum number of points to form a polygon (e.g., a triangle) is 3.
- The relationship between `Polygon` and `Point` is 1-to-many. That is, one `Polygon` has many `Points`.
- A `Point` can belong to one or more `Polygons` (this can be depicted with a many-to-many relationship if required).

Sub Que2: What is the smallest number of points required to construct a polygon?

The smallest number of points required to construct a polygon is **3** (for a triangle).

Sub Que3: Does it make a difference whether or not a point may be shared between polygons?

If points can be shared between polygons, a **`Point`** can be associated with multiple polygons, leading to a many-to-many relationship between **`Point`** and **`Polygon`**.
The ordering of points is essential as it defines the polygon's shape.

Q.3 Figure 3 is a partially completed class diagram of an air transportation system. Add multiplicities in the diagram. Also, add association names to unlevelled associations.

Answer:

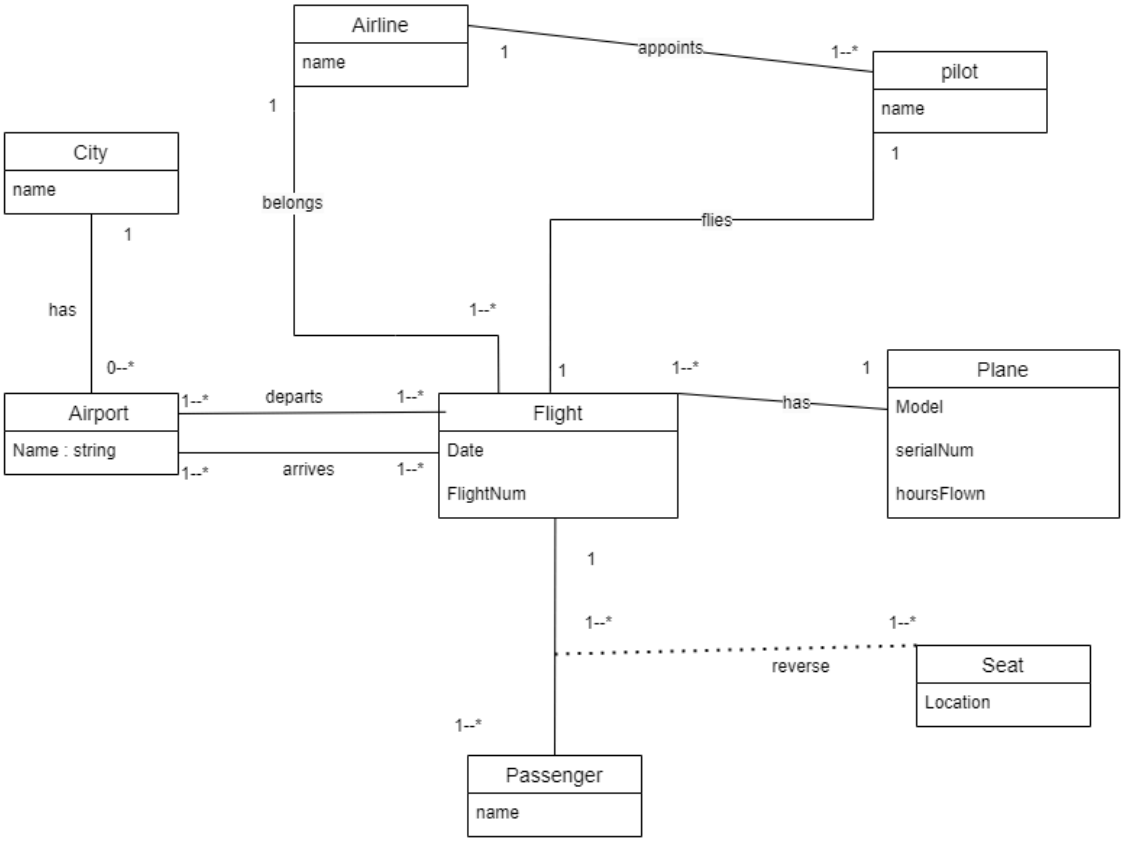
Add Association Names:

- **City to Airport**: A `City` has one or more `Airports`.
- **Airport to Flight**: A `Flight` departs from or arrives at an `Airport`.
- **Airline to Flight**: An `Airline` operates multiple `Flights`.
- **Airline to Pilot**: An `Airline` employs multiple `Pilots`.
- **Pilot to Flight**: A `Pilot` pilots one or more `Flights`.
- **Flight to Plane**: A `Flight` uses one `Plane`.
- **Plane to Seat**: A `Plane` has multiple `Seats`.
- **Flight to Seat**: A `Flight` contains multiple `Seats`.
- **Seat to Passenger**: A `Seat` is occupied by one `Passenger`.

Completed Diagram:

1. **City** `0..*` —— `1` **Airport** (`Has`)
2. **Airport** `1` —— `0..*` **Flight** (`Depart`, `Arrive`)
3. **Airline** `1` —— `0..*` **Flight** (implicitly related via `Operates`)
4. **Airline** `1` —— `0..*` **Pilot** (`Employs`)
5. **Pilot** `0..*` —— `1..*` **Flight** (implicitly related via `Pilots`)
6. **Flight** `1` —— `1` **Plane** (`Uses`)
7. **Plane** `1` —— `1..*` **Seat** (`HasSeats`)
8. **Flight** `1` —— `1..*` **Seat** (`Contains`)
9. **Seat** `1` —— `0..1` **Passenger** (`OccupiedBy`)

Question: 3


Airline
name
appoints
1
1--*
pilot
name
1
1
City
name
belongs
flies
1
has
1--*
0--*
1
1--*
1
Plane
Model
serialNum
hoursFlown
1--*
departs
1--*
has
Airport
Name : string
1--*
arrives
1--*
Flight
Date
FlightNum
1
1--*
1--*
reverse
Seat
Location
1--*
Passenger
name

Q.4 We want to model a system for the management of flights and pilots. An airline operates flights. Each airline has an ID. Each flight has an ID, a departure airport and an arrival airport: an airport as a unique identifier. Each flight has a pilot and a co-pilot, and it uses an aircraft of a certain type; a flight also has a departure time and an arrival time. An airline owns a set of aircraft of different types. An aircraft can be in a working state or it can be under repair. In a particular moment, an aircraft can be landed or airborne. A company has a set of pilots: each pilot has an experience level: 1 is minimum, 3 is maximum. A type of airplane may need a particular number of pilots, with a different role (e.g.: captain, co-pilot, navigator): there must be at least one captain and one co-pilot, and a captain must have a level 3.

Answer: