# Lab04 – Assignment

## IT-314 – Software engineering


## Jal Dani

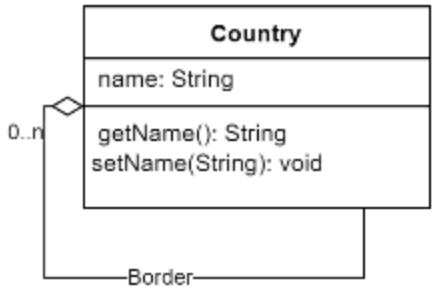
## 202201315


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


Figure-1

- Class Name: Country
- Attributes
  - "name": String
- Association
  - Borders: A many-to-many association between the Countryside

Question:1




Each Country (Spain, France, Belgium) is an instance of the country class.

This class diagram generalizes the structure shown in your object diagram by abstracting individual countries

## Q2

**Prepare a class diagram for object diagram given in Figure -2. Explain your multiplicity decisions. What is the smallest number of points required to construct a polygon? Does it make a difference whether or not points may be shared between polygons? Your answer should address the fact that points are ordered.**

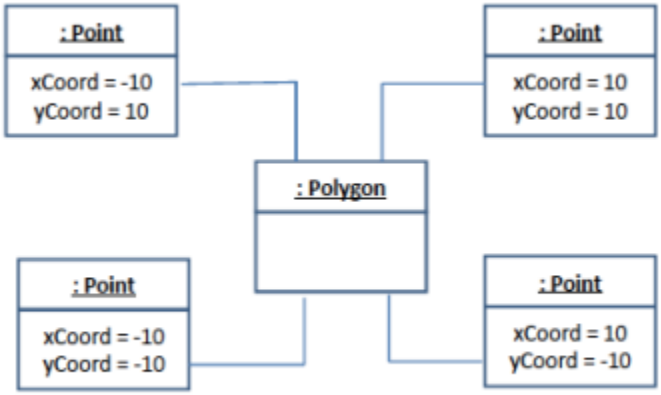



Figure - 2

**Classes:**

- **Point**
- **Polygon**

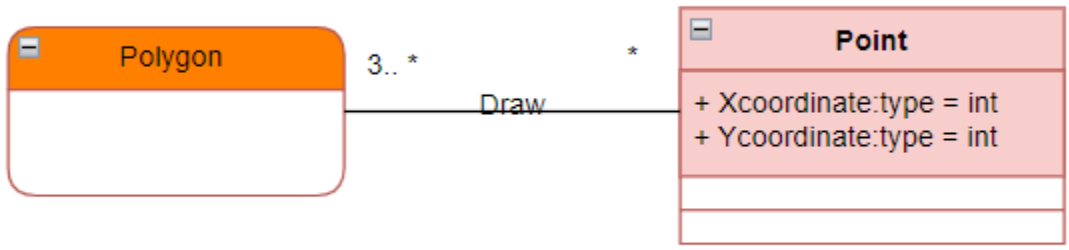
**Attributes:**

- **Point Class:**
  - **Xcoord: int**
  - **Ycoord: int**
- **Polygon:**
  - **The polygon class itself does not seem to have additional attributes.**

## Relationship

- **Polygon and Point**
  - **A polygon has a one-to-many relation.**
  - **Multiplicity: At least 3 needed.**

**Class Diagram Representation**




## Explanation of Multiplicity:

**Polygon to Point:**

A polygon has at least 3 points (hence 1...* cardinality).

A point can belong to multiple polygons (many-to-many relationship).

**Smallest Number of Points:**

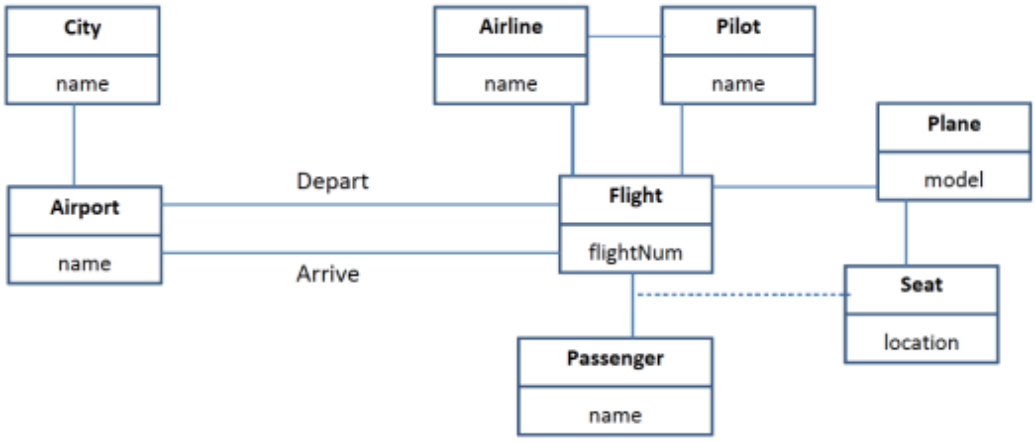
The smallest number of points required to construct a polygon is 3, as a polygon needs at least three vertices to form a closed shape (a triangle).

**Sharing Points Between Polygons:**

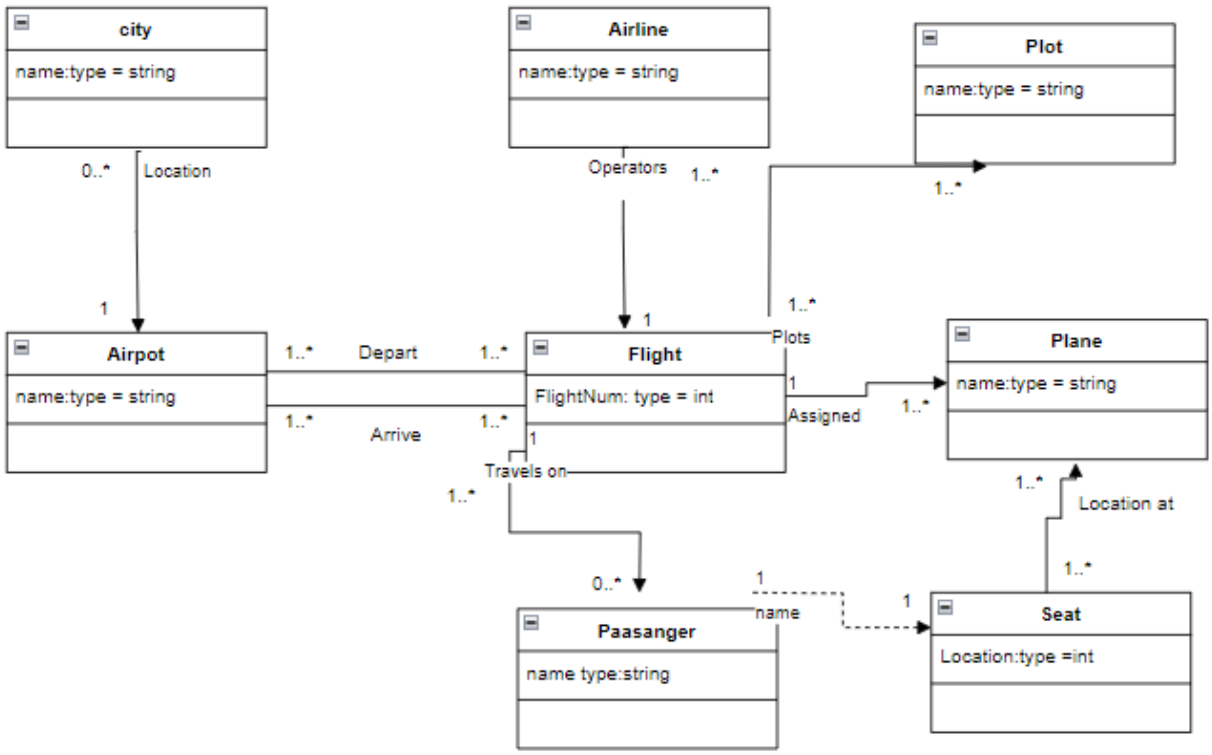
Yes, points can be shared between polygons (e.g., neighboring polygons on a mesh or grid may share vertices). This does not affect the validity of the polygon, but it may impact the specific shape depending on how points are shared and ordered.

## Q3

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

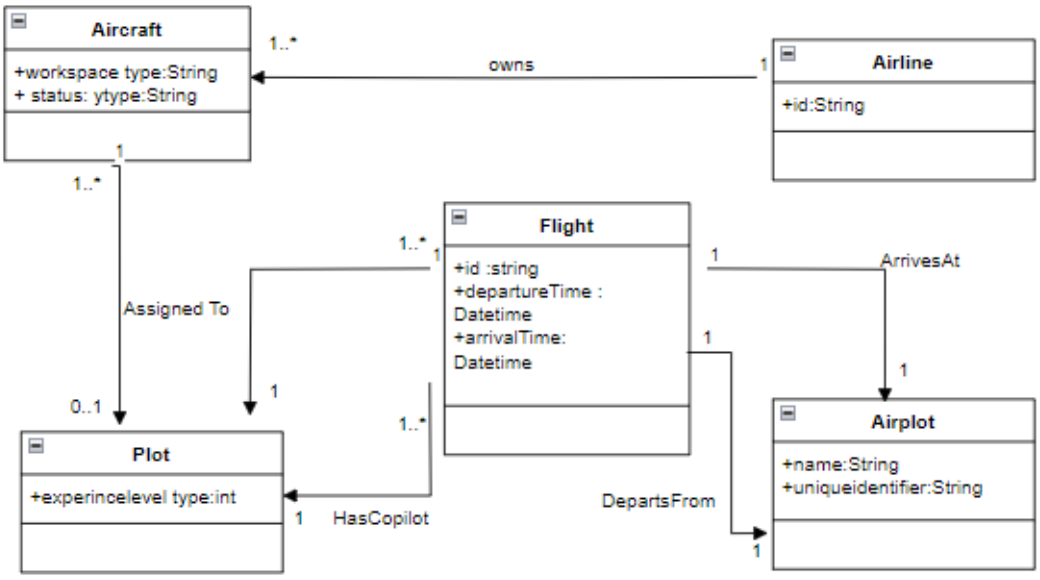


Class Diagram Representation:



**Q4**

We want to model a system for 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 has also 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.: ca0ptain, co-pilot, navigator): there must be at least one captain and one co-pilot, and a captain must have a level 3.


Aircraft
+workspace type:String
+ status: ytype:String
1..*
owns
1
Airline
+id:String
1
1..*
Assigned To
0..1
Flight
+id :string
+departureTime :
Datetime
+arrivalTime:
Datetime
1..*
1
1
1..*
1
ArrivesAt
1
Plot
+experincelevel type:int
1
HasCopilot
DepartsFrom
1
Airplot
+name:String
+uniqueidentifier:String