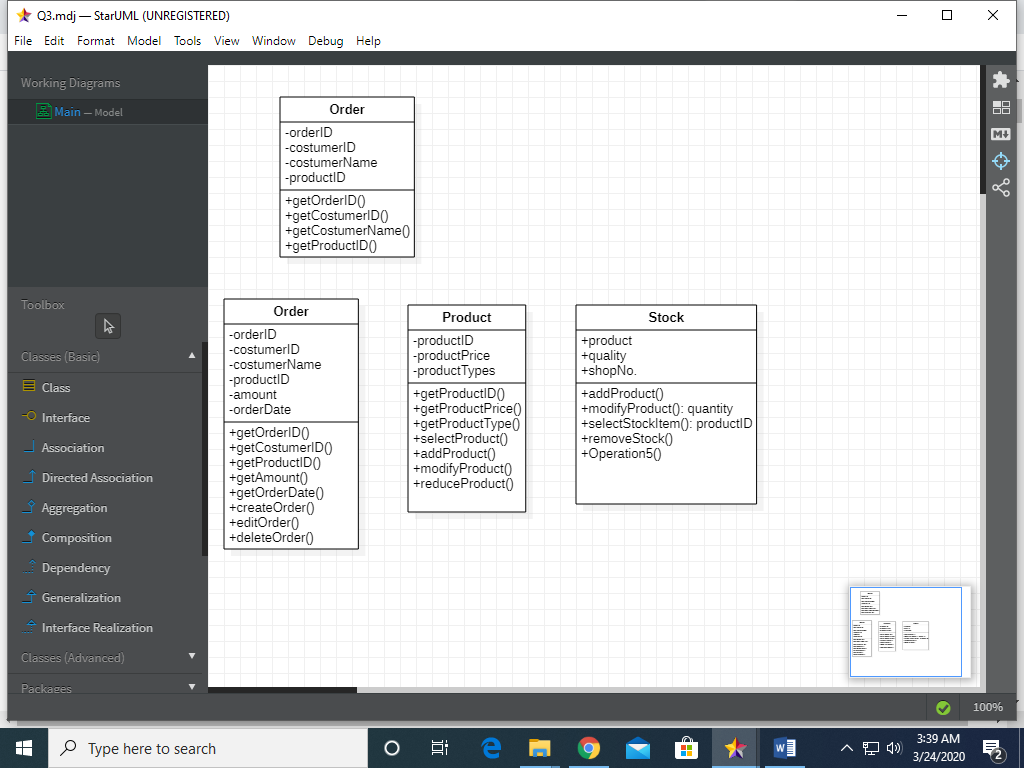
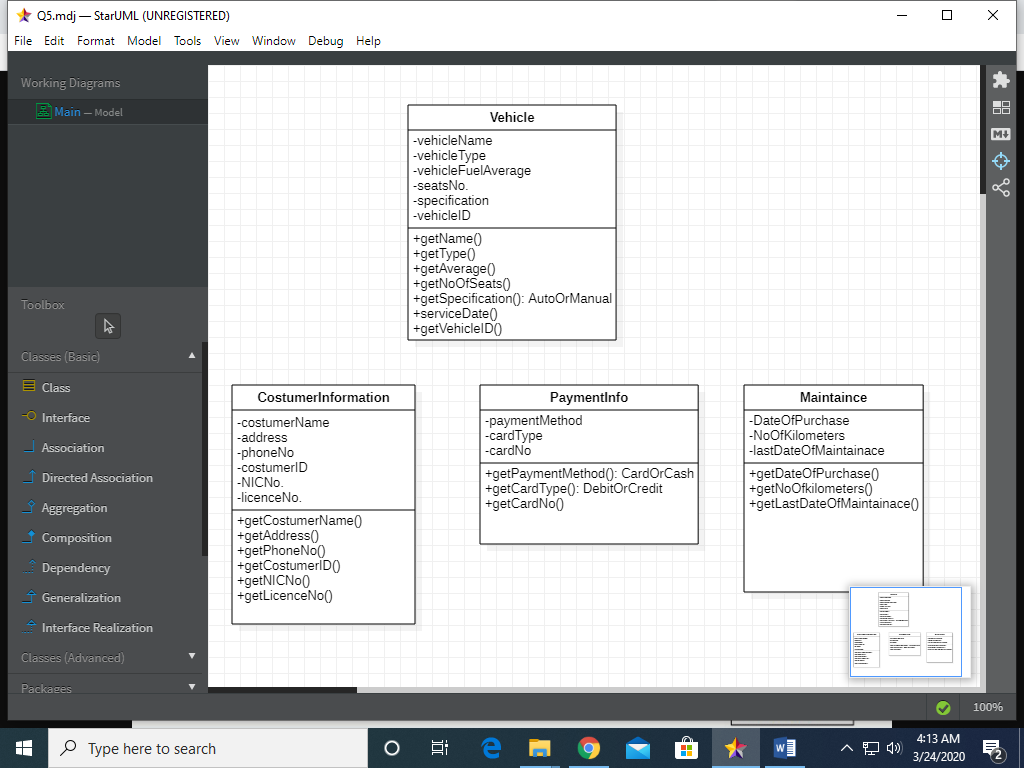


Question 01

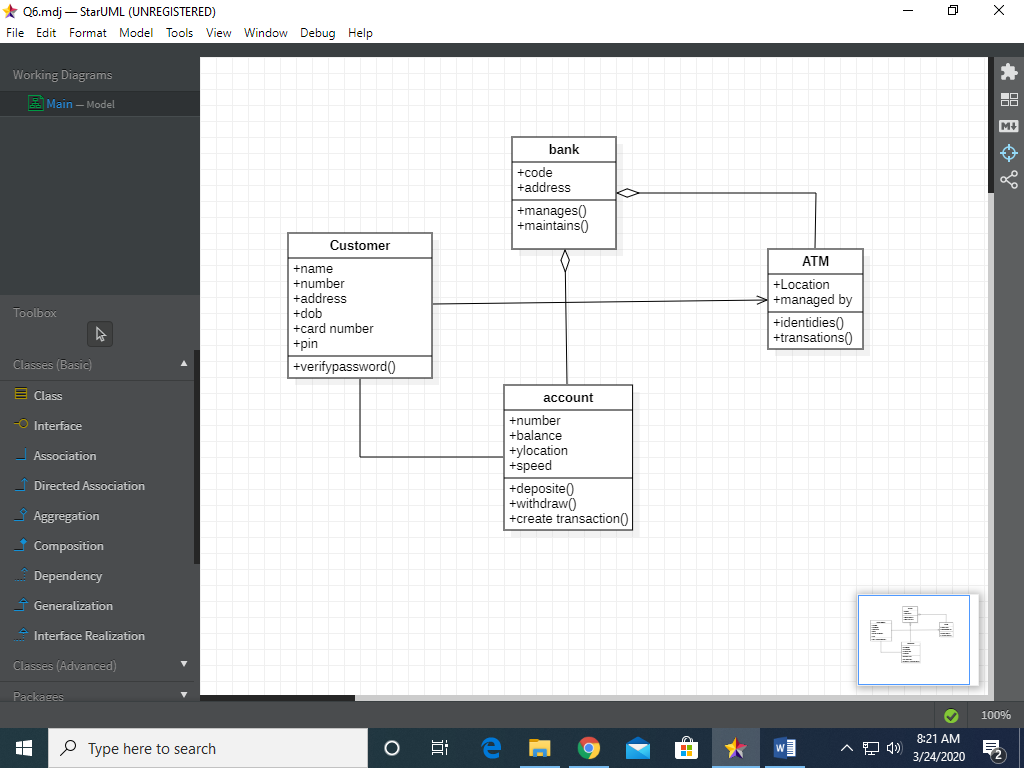
Question 02



Question 03



Question 05

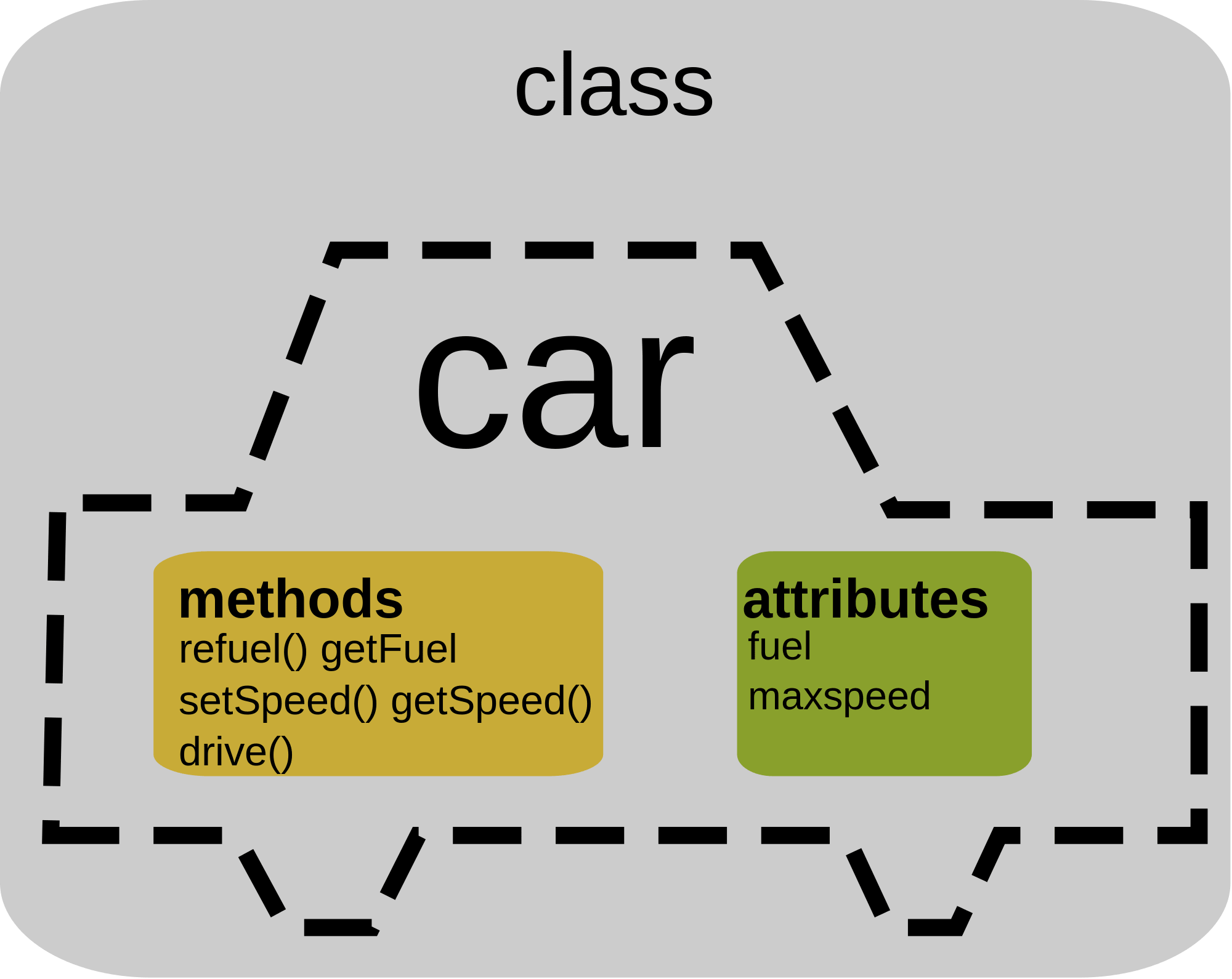


Question 07

Question 06

**Classes:**

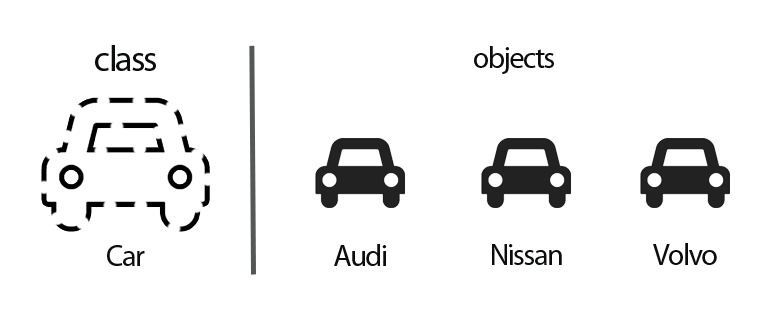
**Classes (OOP) In object-oriented programming, a class is a blueprint for creating objects (a particular data structure), providing initial values for state (member variables or attributes), and implementations of behavior (member functions or methods). The user-defined objects are created using the class keyword.**

**Example:**

**OBJECTS :**

**An Objects is the instance of the class.**

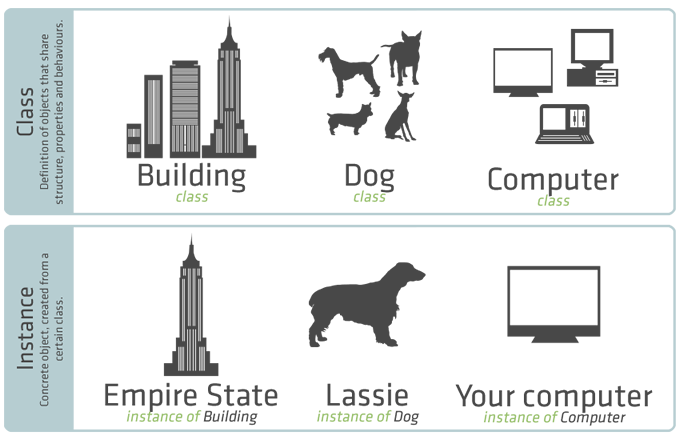
**Example:**

****

**Instances:**

**An instance, in object-oriented programming (OOP), is a specific realization of any object. An object may be varied in a number of ways. ... Each time a program runs, it is an instance of that program. In languages that create objects from classes, an object is an instantiation of a class.**

**Example:**

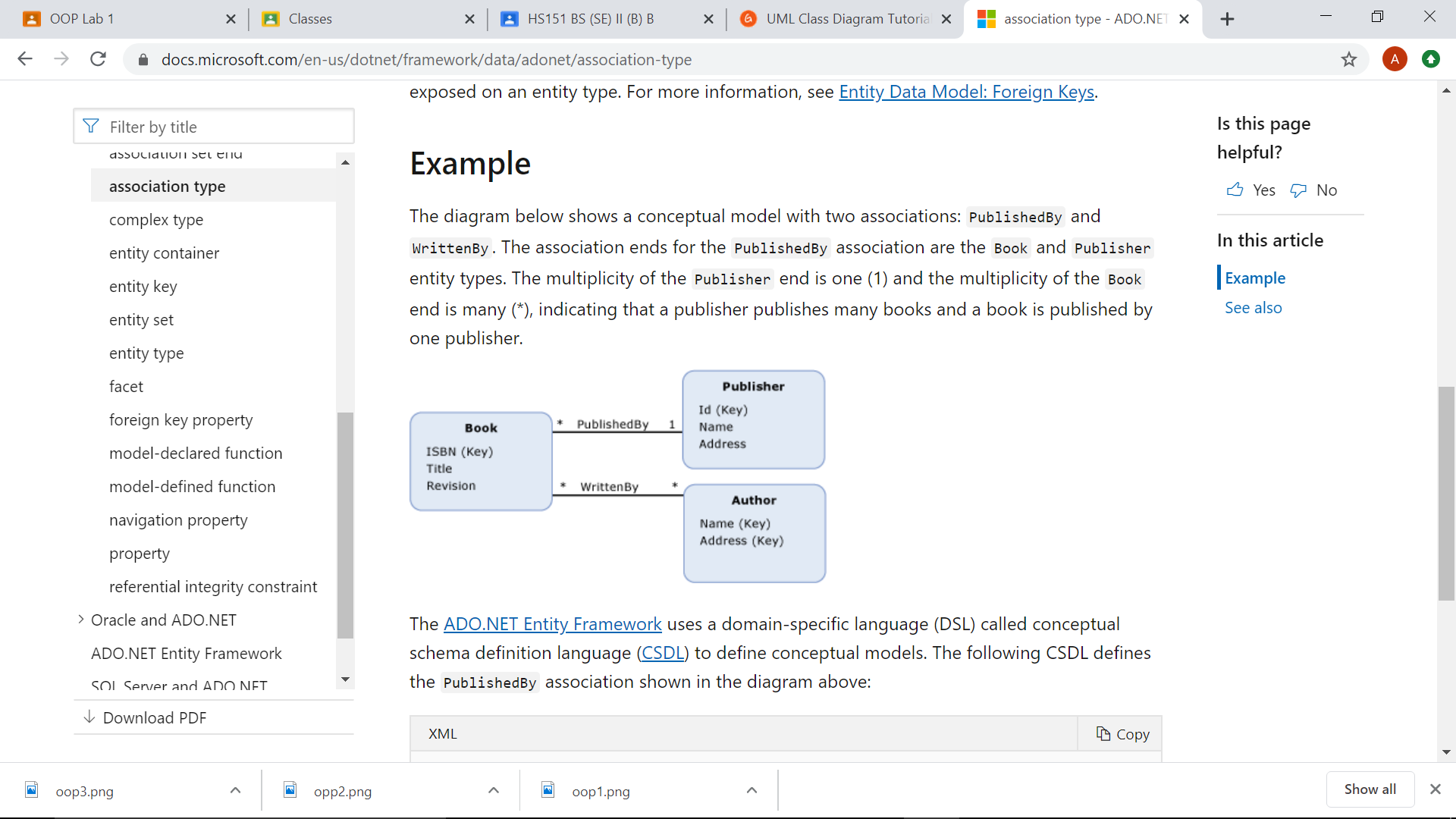
****

**Associations:**

**An association type (also called an association) is the fundamental building block for describing relationships in the Entity Data Model (EDM). In a conceptual model, an association represents a relationship between two entity types (such as Customer and Order )**

**Example:**

**The diagram below shows a conceptual model with two associations: PublishedBy and WrittenBy. The association ends for the PublishedBy association are the Book and Publisher entity types. The multiplicity of the Publisher end is one (1) and the multiplicity of the Book end is many (\*), indicating that a publisher publishes many books and a book is published by one publisher.**

****

# **Access Modifiers in Python**

**In most of the object-oriented languages access modifiers are used to limit the access to the variables and functions of a class. Most of the languages use three types of access modifiers, they are - private, public and protected.**

**Uses:**

**The members declared as Public are accessible from outside the Class through an object of the class.**

**The members declared as Protected are accessible from outside the class but only in a class derived from it that is in the child or subclass.**

**These members are only accessible from within the class. No outside Access is allowed.**