Knowledge Test K7

COMP 4081 • Software Engineering • Fall 2019

Solutions

Name: _____, _____
Last name First name

Rules:

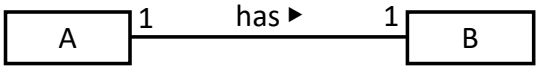
- No potty breaks.
- Turn off cell phones/devices.
- Closed book, closed note, closed neighbor.

- WEIRD! Do not write on the backs of pages. If you need more pages, ask me for some.

Reminders:

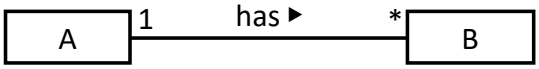
- Verify that you have all pages.
- Don't forget to write your name.
- Read each question carefully.
- Don't forget to answer every question.

For each of the following diagrams, circle the <u>two</u> answers that correctly express the association relationship depicted.



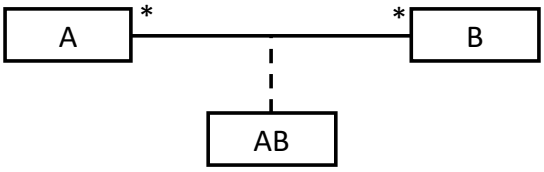
1. [1]

a) Each A has one B
b) Each A has many Bs
c) Each A belongs to one B
d) Each B has one A
e) Each B has many As
f) Each B belongs to one A



2. [1]

a) Each A has one B
b) Each A has many Bs
c) Each A belongs to one B
d) Each B has one A
e) Each B has many As
f) Each B belongs to one A

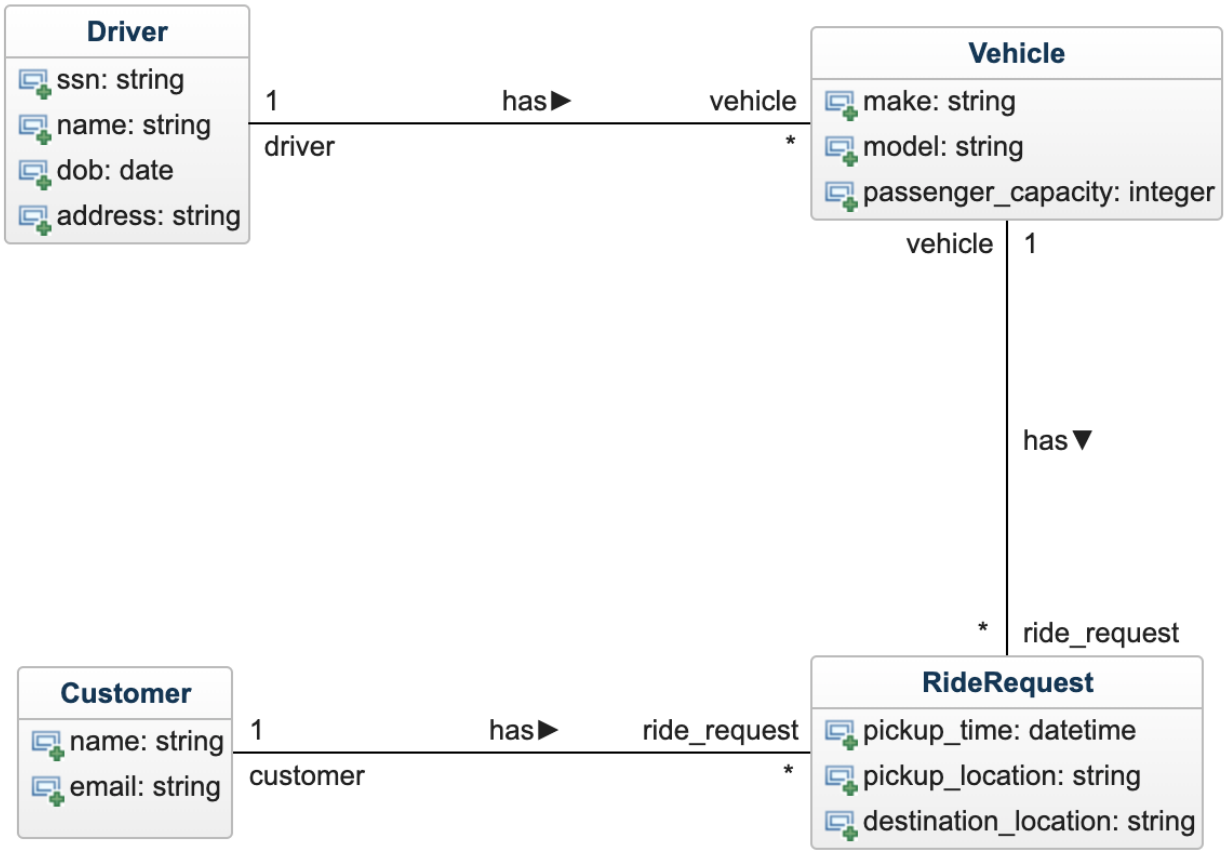


3. [1]

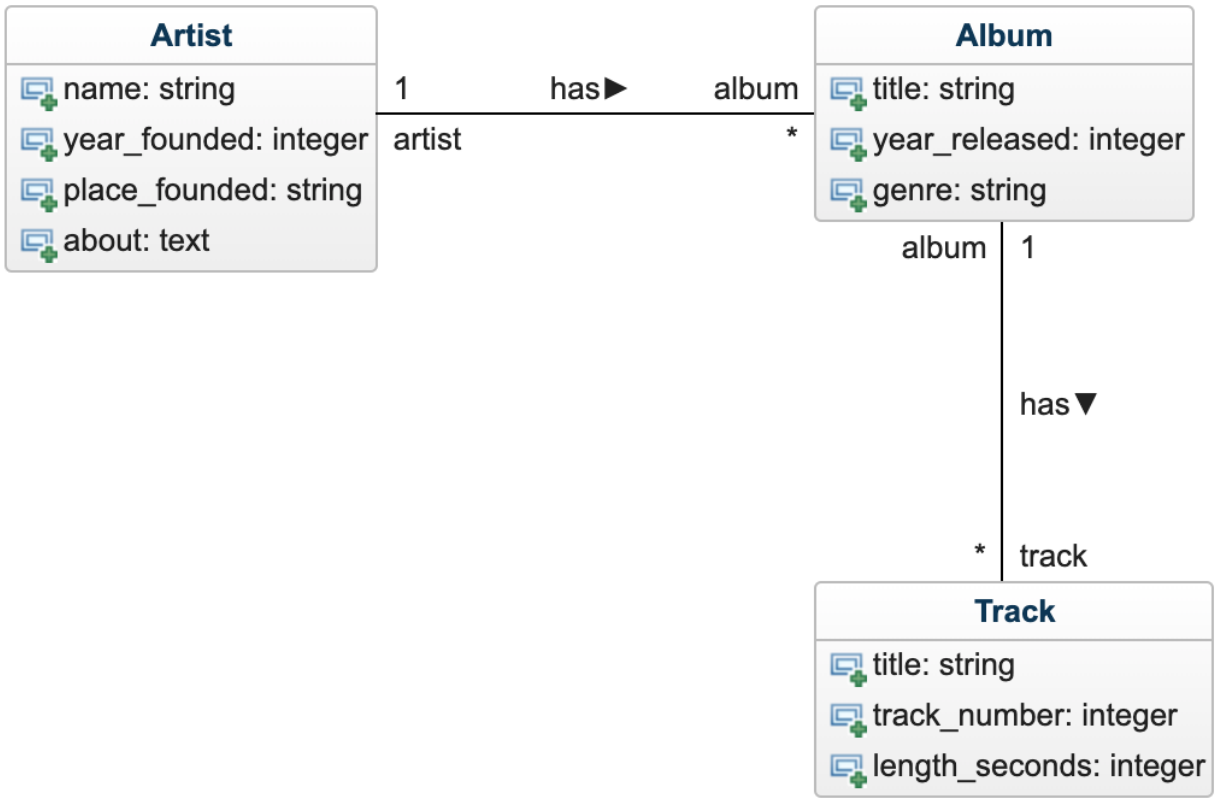
a) Each A has one B
b) Each A has many Bs
c) Each B has one A
d) Each B has many As

4. You have been asked to build a taxicab system similar to Uber. Create an object-oriented data model based on the following natural-language requirements. When deciding what to include, remember that the point here is that you are creating a design for your Rails MVC model. Your answer should take the form of a UML class diagram. Include only things that are specifically described.
   a) [6] Include all relevant classes and attributes.
   b) [6] Include all relevant associations and generalization relationships. Label all associations and association ends and include all multiplicities.

   A driver can register one or more of their vehicles with the system. Vehicles have a make, model, and passenger capacity. A driver has some personal information including their SSN, name, date of birth, and address. Customers have a name and email. Customers can submit a request for a ride with a specific vehicle. In the request, the customer specifies the pick-up time and location (an address) and the destination location address.



5. Draw a UML class diagram that represents the three model classes given in Figure 1.
   a) [6] Include all relevant classes and attributes. Don't include any "id" attributes (including foreign keys). You may also omit the "datetime" attributes that Rails provides by default.
   b) [6] Include all relevant associations and generalization relationships. Label all associations and association ends and include all multiplicities.



6. [1] Which of the following is meant by a *software engineering process*?

   a) A running instance of a program; for example, a UNIX process is a software engineering process

   b) An organization or structure imposed on the tasks and activities involved in developing a software product; for example, developing iteratively and incorporating best practices might be ingredients in a software engineering process

   c) Something developers use to accomplish a goal during a project; for example, Git or Subversion is a software engineering process for configuration management

7. [1] In the _____ development process, development of a system proceeds through repeated cycles and in smaller portions at a time, allowing software developers to take advantage of what was learned during development of earlier parts or versions of the system.

   a) iterative

   b) verified

   c) waterfall

8. [1] In the _____ development process, the various phases of development are completed sequentially, one after the other (e.g., gather all the requirements, then design the whole system, then implement the whole system, and so on).

   a) iterative

   b) verified

   c) waterfall

Bonus Problems

9. [1] What often-false assumption does the *waterfall process model* made about requirements specifications?

   a) Specifications are predictable

   b) Specifications are stable

   c) Specifications have low change rates

   d) All of the above

   e) None of the above

10. [1] An *empirical process model* iterates between...

    a) ... design and implementation

    b) ... requirements gathering and design

    c) ... feedback and adaptation

11. [1] In iterative development, how long should an iteration generally be?

    a) 1 week

    b) 2-6 weeks

    c) 2-4 months

12. [1] In software engineering, defects that are discovered _____ are generally _____ to fix.

    a) earlier; more expensive

    b) later; more expensive

    c) by customers; more difficult

    d) by developers; more difficult

13. [1] Following a(n) _____ software engineering process tends to reveal defects early in development.

    a) iterative

    b) sequential

    c) waterfall

14. [1] All else being equal, choose the estimate below that is most likely to be accurate.

    a) 1 day

    b) 1 week

    c) 1 month

15. [4] What two things are wrong with the following series of steps?

First, the developers solicit user stories from the customer.

Next, the developers assign a priority level to each user story.

Next, the developers estimate the effort required to implement each user story.

(1) It is the customers who assign priority levels, not the developers. Only the customers know their priorities.
(2) The second and third steps are in the wrong order. Estimates must be done before priorities. In setting priorities, customers must have a sense of the cost and benefit of each feature. The estimates tell them the cost.

Figures

```
# == Schema Information
#
# Table name: artists
#
#  id             :integer          not null, primary key
#  name           :string
#  year_founded   :integer
#  place_founded  :string
#  about          :text
#  created_at     :datetime         not null
#  updated_at     :datetime         not null
#

class Artist < ApplicationRecord
    has_many :albums
    validates :year_founded, numericality: { less_than_or_equal_to: Date.today.year }
end
```

```
# == Schema Information
#
# Table name: albums
#
#  id             :integer          not null, primary key
#  title          :string
#  year_released  :integer
#  genre          :string
#  artist_id      :integer
#  created_at     :datetime         not null
#  updated_at     :datetime         not null
#
# Indexes
#
#  index_albums_on_artist_id  (artist_id)
#

class Album < ApplicationRecord
  belongs_to :artist
  has_many :tracks
  validates :genre, inclusion: { in: ['Rock', 'R&B/HipHop', 'Pop', 'Country', 'Latin'] }
end
```

```
# == Schema Information
#
# Table name: tracks
#
#  id              :integer          not null, primary key
#  title           :string
#  track_number    :integer
#  length_seconds  :integer
#  album_id        :integer
#  created_at      :datetime         not null
#  updated_at      :datetime         not null
#
# Indexes
#
#  index_tracks_on_album_id  (album_id)
#

class Track < ApplicationRecord
  belongs_to :album
end
```

Figure 1. Model classes for a music catalog application.