Interview questions - Brain teasers

Many companies, including Google and Microsoft, have policies banning brain teasers, still interviewers sometimes ask these tricky questions since people have different definitions of brain teasers.

1. Use arithmetic operations (+, -, ×, ÷) and parentheses to make the statement 3 1 3 6 = 8 true

Interview questions - Brain teasers

Many companies, including Google and Microsoft, have policies banning brain teasers, still interviewers sometimes ask these tricky questions since people have different definitions of brain teasers.

1. Use arithmetic operations (+, -, ×, ÷) and parentheses to make the statement 3 1 3 6 = 8 true

a) assume that 6 is outside parentheses, hence

$$3 \quad 1 \quad 3 \;\; \boxed{\begin{matrix} + \\ - \\ \times \\ \div \end{matrix}} \;\; 6 = 8$$

Interview questions - Brain teasers

Many companies, including Google and Microsoft, have policies banning brain teasers, still interviewers sometimes ask these tricky questions since people have different definitions of brain teasers.

1. Use arithmetic operations (+, -, ×, ÷) and parentheses to make the statement 3 1 3 6 = 8 true

a) assume that 6 is outside parentheses, hence

$$3\quad 1\quad 3\ \boxed{\begin{matrix}+\\-\\\times\\\div\end{matrix}}\ 6 = 8$$

3 1 3 + 6 = 8
3 1 3 = 2 nope

Interview questions - Brain teasers

Many companies, including Google and Microsoft, have policies banning brain teasers, still interviewers sometimes ask these tricky questions since people have different definitions of brain teasers.

1. Use arithmetic operations (+, -, ×, ÷) and parentheses to make the statement 3 1 3 6 = 8 true

a) assume that 6 is outside parentheses, hence

3 1 3 + 6 = 8

- (-, ×, ÷)

3 1 3 - 6 = 8

3 1 3 = 14

nope

Interview questions – Brain teasers

Many companies, including Google and Microsoft, have policies banning brain teasers, still interviewers sometimes ask these tricky questions since people have different definitions of brain teasers.

1. Use arithmetic operations (+, -, ×, ÷) and parentheses to make the statement 3 1 3 6 = 8 true

a) assume that 6 is outside parentheses, hence

3 1 3 + 6 = 8
-
×
÷

3 1 3 × 6 = 8

3 1 3 = 4/3 YES!

(3 + 1) ÷ 3 = 4/3

Answer: (3 + 1) ÷ 3 × 6 = 8

Interview questions - OOD

Object oriented design questions are very important, as they demonstrate the quality of a candidate’s code

OOD questions are often intentionally vague to test if you’ll make assumptions, or if you’ll ask clarifying questions.

How do you design a class if the constraints are vague? Ask questions to eliminate ambiguity, then design the classes to handle any remaining ambiguity.

1. Design the data structures for a generic deck of cards. Explain how you would subclass it to implement particular card games.

Interview questions - OOD

1. Design the data structures for a generic deck of cards. Explain how you would subclass it to implement particular card games.

Consider the following questions:

A. What are you trying to do with the deck of cards? *Ask your interviewer.*

- we want a general purpose deck of cards to implement many different types of card games

Interview questions - OOD

1. Design the data structures for a generic deck of cards. Explain how you would subclass it to implement particular card games.

Consider the following questions:

A. What are you trying to do with the deck of cards? *Ask your interviewer.*

- we want a general purpose deck of cards to implement many different types of card games

B. What are the core objects—and what “sub types” are there?

Examples: Card, Deck, Number, Suit, PointValue

Interview questions - OOD

1. Design the data structures for a generic deck of cards. Explain how you would subclass it to implement particular card games.

Consider the following questions:

C. Missed anything?

- think about how you'll use that deck of cards to implement different types of games, changing the class design as necessary.

Interview questions - OOD

1. Design the data structures for a generic deck of cards. Explain how you would subclass it to implement particular card games.

Consider the following questions:

C. Missed anything?

- think about how you'll use that deck of cards to implement different types of games, changing the class design as necessary.

D. get a little deeper: how will the methods work?

If you have a method like `Deck.getCard(Suit s, Number n)`, think about how it will retrieve the card.

Interview questions - OOD

1. Design the data structures for a generic deck of cards. Explain how you would subclass it to implement particular card games.

Consider the following questions:

C. Missed anything?

- think about how you'll use that deck of cards to implement different types of games, changing the class design as necessary.

D. get a little deeper: how will the methods work?

If you have a method like `Deck.getCard(Suit s, Number n)`, think about how it will retrieve the card.

We will develop some of these classes in CSI33

Interview questions - OOD

2. Imagine you have a call center with three levels of employees: fresher, technical lead (TL), and product manager (PM). There can be multiple employees, but only one TL or PM. An incoming telephone call must be allocated to a fresher who is free. If a fresher can't handle the call, he or she must escalate the call to technical lead. If the TL is not free or not able to handle it, then the call should be escalated to PM.

Design the classes and data structures for this problem. Implement a method getCallHandler().

Interview questions - OOD

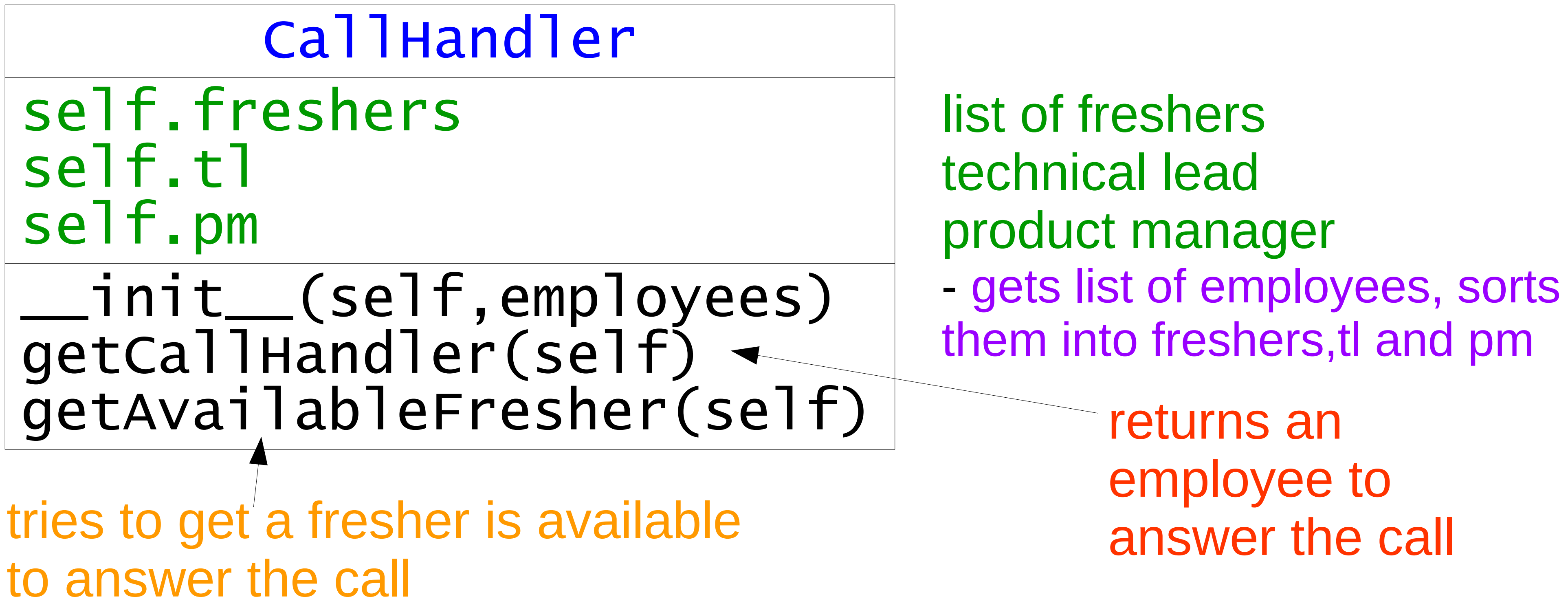
2. employees: fresher, technical lead (TL), product manager (PM).
An incoming telephone → fresher → TL → PM
method `getCallHandler()`

We assume that `getCallHandler` should return Employee instance.

Interview questions - OOD

2. employees: fresher, technical lead (TL), product manager (PM).
An incoming telephone → fresher → TL → PM
method `getCallHandler()`

We assume that `getCallHandler` should return Employee instance.



CallHandler
self.freshers self.tl self.pm

__init__(self,employees)
getCallHandler(self)
getAvailableFresher(self)

list of freshers
technical lead
product manager
- gets list of employees, sorts them into freshers,tl and pm

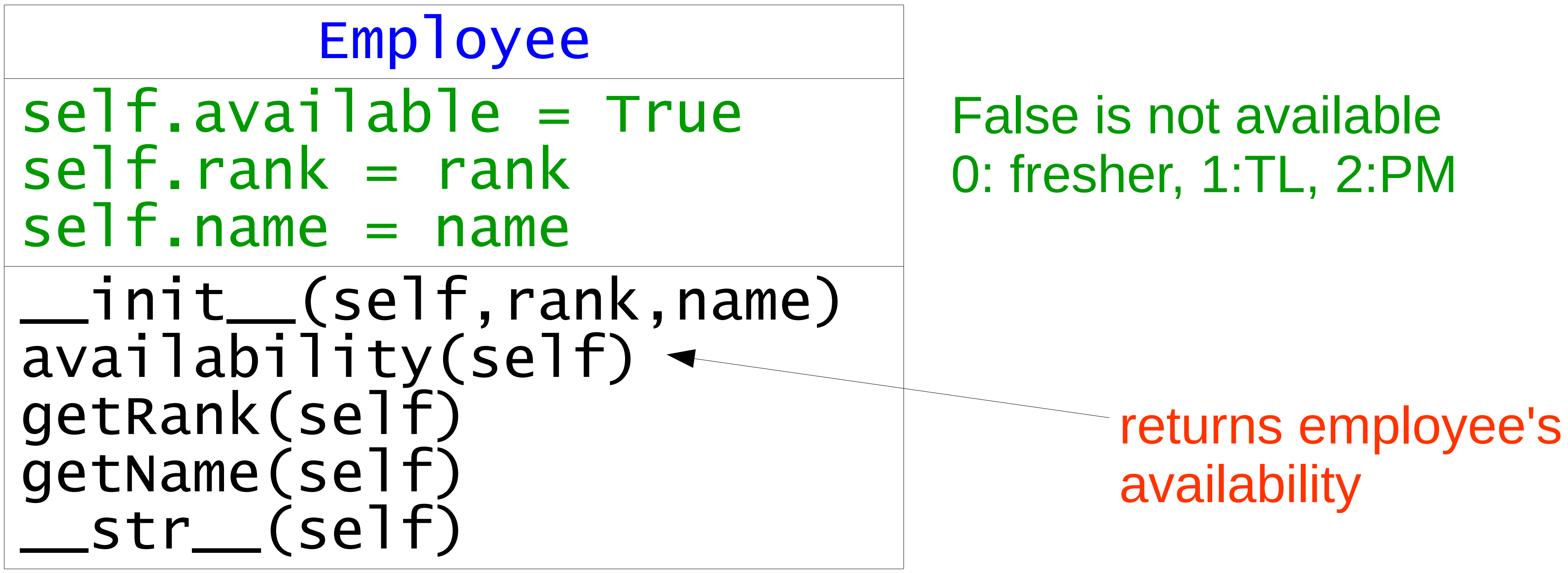
returns an employee to answer the call

tries to get a fresher is available to answer the call

Interview questions - OOD

2. employees: fresher, technical lead (TL), product manager (PM).
An incoming telephone → fresher → TL → PM
method `getCallHandler()`

We assume that `getCallHandler` should return Employee instance.



```
Employee
-----------------------------
self.available = True
self.rank = rank
self.name = name
-----------------------------
__init__(self,rank,name)
availability(self)
getRank(self)
getName(self)
__str__(self)
```

False is not available
0: fresher, 1:TL, 2:PM

returns employee's availability

Interview questions - OOD

2. employees: fresher, technical lead (TL), product manager (PM).
An incoming telephone → fresher → TL → PM
method `getCallHandler()`

We assume that `getCallHandler` should return Employee instance.

See program callHandler.py

Interview questions - OOD

3. Explain how you would design a chat server. In particular, provide details about the various backend components, classes, and methods. What would be the hardest problems to solve?

Interview questions - OOD

3. Explain how you would design a chat server. In particular, provide details about the various backend components, classes, and methods. What would be the hardest problems to solve?
- this is something we played with during our recent meetings

A. discuss with your interviewer “What is our chat server?”

Interview questions - OOD

3. Explain how you would design a chat server. In particular, provide details about the various backend components, classes, and methods. What would be the hardest problems to solve?
- this is something we played with during our recent meetings

A. discuss with your interviewer “What is our chat server?”

B. Our assumptions: a basic chat server that needs to support a small number of users.

People have a contact list, they see who is online vs offline, and they can send text-based messages to them.

We will not worry about supporting group chat, voice chat, etc.

We will also assume that contact lists are mutual: I can only talk to you if you can talk to me. Let’s keep it simple.

Interview questions - OOD

3. Explain how you would design a chat server. In particular, provide details about the various backend components, classes, and methods. What would be the hardest problems to solve?

What specific actions does it need to support?
>> User A signs online
>> User A asks for their contact list, with each person's current status.
>> Friends of User A now see User A as online
>> User A adds User B to contact list
>> User A sends text-based message to User B
>> User A changes status message and/or status type
>> User A removes User B
>> User A signs offline

Interview questions - OOD

3. Explain how you would design a chat server. In particular, provide details about the various backend components, classes, and methods. What would be the hardest problems to solve?

What can we learn about these requirements?
need a concept of users,
add request status,
online status, and
messages.

What are the core components?
- a database to store items,
- an “always online” application as the server
- might recommend using XML for the communication between the chat server and the clients, as it’s easy for a person and a machine to read.

Interview questions - OOD

3. Explain how you would design a chat server. In particular, provide details about the various backend components, classes, and methods. What would be the hardest problems to solve?

A list of key objects and methods (many details are hidden):

```
StatusType = ("online","offline","away")

class User:
    self.username
    self.display_name
   self.contact_list
   self.requests
bool def updateStatus(self,StatusType,message) T/F
bool def addUserWithUsername(self,name)  T/F
bool def approveRequest(self,username)  T/F
bool def denyRequest(self,username) T/F
bool def removeContact(self.username) T/F
bool def sendMessage(self,username,message) T/F
```

Interview questions - OOD

3. Explain how you would design a chat server. In particular, provide details about the various backend components, classes, and methods. What would be the hardest problems to solve?

A list of key objects and methods (many details are hidden):

```
class AddRequest:
   self.from_user
   self.to_user

class Server:
   def __init__(self,...)
   def getUserByUsername(self,username)
   ...
```

Interview questions - OOD

3. Explain how you would design a chat server. In particular, provide details about the various backend components, classes, and methods. What would be the hardest problems to solve?

What problems would be the hardest to solve (or the most interesting)?

1) How do we know if someone is online?

while we would like users to tell us when they sign off, we can’t know for sure.
A user’s connection might have died, for example.
To make sure that we know when a user has signed off, we might try regularly pinging the client to make sure it’s still there.

Interview questions - OOD

3. Explain how you would design a chat server. In particular, provide details about the various backend components, classes, and methods. What would be the hardest problems to solve?

What problems would be the hardest to solve (or the most interesting)?

2) How do we deal with conflicting information?

We have some information stored in the computer's memory and some in the database.
What happens if they get out of sync? Which one is "right"?

Interview questions - OOD

3. Explain how you would design a chat server. In particular, provide details about the various backend components, classes, and methods. What would be the hardest problems to solve?

What problems would be the hardest to solve (or the most interesting)?

2) How do we deal with conflicting information?

We have some information stored in the computer’s memory and some in the database.
What happens if they get out of sync? Which one is “right”?

3) How do we make our server scale?
While we designed out chat server without worrying—too much– about scalability, in real life this would be a concern. We’d need to split our data across many servers, which would increase our concern about out of sync data.

Interview questions - OOD

3. Explain how you would design a chat server. In particular, provide details about the various backend components, classes, and methods. What would be the hardest problems to solve?

What problems would be the hardest to solve (or the most interesting)?

4) How we do prevent denial of service attacks?
Clients can push data to us—what if they try to DOS us? How do we prevent that?

Interview questions - OOD

4. Othello is played as follows: Each Othello piece is white on one side and black on the other. When a piece is surrounded by its opponents on both the left and right sides, or both the top and bottom, it is said to be captured and its color is flipped. On your turn, you must capture at least one of your opponent’s pieces. The game ends when either user has no more valid moves, and the win is assigned to the person with the most pieces. Implement the object oriented design for Othello.

- one of moderate final projects