Northeastern University

CS 2100: Program Design and Implementation 1

Practice Quiz 2

Instructions

- Please put all of your answers on the answer sheet. Only the answer sheet will be graded.
- Do not begin the quiz until instructed to do so.
- You may use both sides of a sheet of paper up to 8.5"x11" for reference, but no other resources, including phones, computers, AI, headphones, and ear pods.
- You have until the end of the class period to complete the quiz.
- Students may not leave the classroom during the first 10 minutes of the quiz (except in case of emergency).
- Hand your completed answer sheet to an instructor before leaving the room.
- Talk to an instructor if you need to leave the room and reenter.

Please use the following code to answer the questions below:

```
class Pillow:
     """A pillow with a cover"""

     def init(self, pillow_cover_type: str) -> None:
          self.cover_type = pillow_cover_type

     def __str__(self) -> str:
          return f'{self.cover_type} pillow'

class Blanket:
     """A blanket with a cover and weight"""

     def init(self, cover_type: str, weight: float) -> None:
          if weight <= 0:
```

```
            raise ValueError('Cannot have negative weight')
        self.cover_type = cover_type
        self.weight = weight

    def __str__(self) -> str:
        return f'{self.weight} pound {self.cover_type} blanket'



class Bed:
    """A bed with pillows and blankets"""

    def init(
            self, width: int, height: int,
            pillows: list[Pillow], blankets: list[Blanket]
    ) -> None:
        self._width = width
        self._height = height
        self.pillows = pillows[:]
        self.blankets = blankets[:]
        self.__sizes = {
            (38, 75): 'twin',
            (38, 80): 'twin XL',
            (54, 75): 'full',
            (60, 80): 'queen',
            (72, 84): 'California king',
            (76, 80): 'king'
        }

    def __str__(self) -> str:
        pillows = ', '.join([str(pillow) for pillow in self.pillows])
        blankets = ', '.join([str(blanket) for blanket in self.blankets])
        return f'{self.size} size mattress, {pillows}, {blankets}'

    @property
    def size(self) -> str:
        """Size of the mattress"""
        return self.__sizes.get((self._width, self._height), 'unknown') # Q9

    @size.setter
    def size(self, new_size: str) -> None:      # Q10
```

```
            for key, value in self.__sizes.items():
                if value == new_size:
                    self._width, self._height = key


def main() -> None:
    my_bed = Bed(
        38, 75,
        [Pillow('silk'), Pillow('silk')],
        [Blanket('velvet', 50)])
    print(my_bed)           # Q4
    print(my_bed.size)      # Q5
    my_bed.size = 'twin XL' # Q6
    print(my_bed._height)   # Q7
    print(my_bed.__sizes)   # Q8

if name == 'main':
    main()
```

Classes: constructors, methods, and attributes

1. What happens if we pass a negative width to the Bed constructor?
    a. It will raise a ValueError
    b. The Bed will be None
    c. It will make the width 0
    d. It will save the negative width to the `_width` attribute as if it was valid
2. Let's say I create a list of pillows called `cat_pillows`. Will `cat_pillows` be empty after the following code?

```
cat_bed = Bed(
        38, 75,
        cat_pillows,
        [Blanket('velvet', 50)])
cat_bed.pillows = []
```

    a. No, because the list `cat_bed.pillows` is a copy of `cat_pillows`
    b. No, because `cat_bed.pillows` cannot be accessed from outside the `Bed` class

c. No, because `cat_bed.pillows` cannot be accessed from outside the `Pillow` class
d. Yes

3. Can there exist a `Blanket` with a negative weight? If so, how?
    a. No
    b. Yes, but only if we take a valid `Blanket` and then modify its `weight` to be negative
    c. Yes, but only if we use a try / except while passing a negative weight to the constructor
    d. Yes, but only if the negative weight is an `int`, not a `float`
4. What is output by the print statement labelled Q4?
    a. twin size mattress, silk pillow, silk pillow, 50 pound velvet blanket
    b. <__main__.Bed object at 0x10cf727e0>
    c. twin size mattress, <__main__.Pillow object at 0x10cb46660>, <__main__.Pillow object at 0x10cb466c0>, <__main__.Blanket object at 0x10cb468d0>
    d. (Nothing)

Properties (attributes with getters and setters)

5. What is output by the print statement labelled Q5?
    a. (38, 75)
    b. twin
    c. It raises an error because we can't access `size` from outside the Bed class
    d. It raises an error because there is no `size` attribute
6. Which of the following statements is true about the statement labelled Q6?
    a. It calls the method with the `@property` decorator
    b. It calls the method with the `@size.setter` decorator
    c. It executes without error, but it does not call either of the methods with decorators
    d. It raises an error
7. What is output by the print statement labelled Q7?
    a. twin XL
    b. twin
    c. 80
    d. 75
8. What is output by the print statement labelled Q8?
    a. my_bed.__sizes
    b. {'twin', 'twin XL', 'full', 'queen', 'California king', 'king'}
    c. {(38, 75): 'twin', (38, 80): 'twin XL', (54, 75): 'full', (60, 80): 'queen', (72, 84): 'California king', (76, 80): 'king'}
    d. It raises an error

Sets and dictionaries

9. In the method `def size(self) -> str` (labelled Q9), what happens if the current values for `self._width` and `self._height` do not correspond to one of the standard mattress sizes?
    a. It returns `'unknown'`
    b. It returns a tuple containing the width and height
    c. It returns `None`
    d. It raises an error
10. In the `size` setter (the method `def size(self, new_size: str) -> None`, labelled Q10), what happens if the client passes a `new_size` which is not one of the standard mattress sizes?
    a. It sets the width and height to 0
    b. It sets the width and height to `None`
    c. It does nothing
    d. It raises an error
11. How can I get a set of all materials used in the Pillows and Blankets of `my_bed`?
    a. `{item.cover_type for item in my_bed.pillows} | {item.cover_type for item in my_bed.blankets}`
    b. `{item.cover_type for item in my_bed.pillows} + {item.cover_type for item in my_bed.blankets}`
    c. `{item.cover_type for item in my_bed.pillows} & {item.cover_type for item in my_bed.blankets}`
    d. `{item.cover_type for item in my_bed.pillows} - {item.cover_type for item in my_bed.blankets}`
12. Can a set of dimensions (a width x height pair) have two different "standard" names for its size? E.g., can `self.__sizes` contain this?
    `{ (10, 20): 'cat', (10, 20): 'bird' }`
    a. Yes, because the values are different
    b. Yes, because the keys are the same
    c. No, because a dictionary cannot have two of the same key
    d. No, because a dictionary cannot have two of the same value

Lists: sorting, mapping, filtering

13. Let's write a method inside the `Bed` class with the signature `def has_material(self, material: str) -> bool` which returns `True` if any of the pillows or blankets contain the specified material. What could be the body of this method?
    a. `return material in (item for item in self.pillows + self.blankets)`
    b. `return material in (item.cover_type for item in self.pillows + self.blankets)`
    c. `return material in (self.pillows + self.blankets)`

    d. `return material in (cover_type for item in self.pillows + self.blankets)`

14. How can we check whether all pillows are covered in either silk or satin? (E.g., there are no pillows covered in anything other than silk or satin)
    a. `all([item in ('silk', 'satin') for item in my_bed.pillows + my_bed.blankets])`
    b. `all([item in ('silk', 'satin') for item.cover_type in my_bed.pillows + my_bed.blankets])`
    c. `all([item in ('silk', 'satin') for item in my_bed.pillows.cover_type + my_bed.blankets.cover_type])`
    d. `all([item.cover_type in ('silk', 'satin') for item in my_bed.pillows + my_bed.blankets])`
15. How can I get a list of blankets, sorted by weight?
    a. `sorted(my_bed.blankets, key=blanket.weight)`
    b. `sorted(my_bed.blankets.weight, key=lambda blanket: blanket)`
    c. `sorted(my_bed.blankets.weight, key=lambda blanket: blanket.weight)`
    d. `sorted(my_bed.blankets, key=lambda blanket: blanket.weight)`
16. How can I get a list of the blankets on `my_bed` which weigh less than 50?
    a. `[blanket for blanket if blanket.weight < 50 in my_bed.blankets]`
    b. `[blanket for blanket in my_bed.blankets if blanket.weight < 50]`
    c. `[blanket if blanket.weight < 50 for blanket in my_bed.blankets]`
    d. `[my_bed.blankets if blanket.weight < 50]`

Stakeholder-value matrices

17. Consider a website where people can book rooms at a hotel. Which of the following is NOT a stakeholder?
    a. Hotel employees
    b. Money
    c. Agents who make hotel bookings for clients
    d. People who cannot walk
18. And, which of *these* is NOT a stakeholder?
    a. Event organizers who want to book many rooms
    b. People who speak a different language than the one local to the hotel
    c. People whose travel plans are tentative
    d. Visual elegance
19. Which of the following is NOT a value?
    a. People's names
    b. Security / fraud detection

c. Fairness
d. Accessibility

20. Out of the following, which one would NOT be appropriate to put in the Stakeholder-Value Matrix cell where the stakeholder is a client making a booking, and the value is financial value?
    a. The client wants to book a room at a reasonable cost.
    b. The client wants to spend more money to justify spending the night in a hotel.
    c. The client would rather pay than look at ads.
    d. The hotel wants to spend as little money as possible building the website.

Correlation

21. Which statement best describes correlation?
    a. A measure of how one variable causes changes in another variable
    b. A measure of the strength and direction of the linear relationship between two variables
    c. The average difference between two variables
    d. A measure of how much one variable varies
22. A Pearson correlation coefficient of r = -0.95 indicates:
    a. A weak negative relationship
    b. A strong negative relationship
    c. Little to no relationship
    d. A measurement error
23. If two variables have a correlation coefficient of approximately 0, this means:
    a. The variables are perfectly related
    b. There is no linear relationship between the variables
    c. One variable causes the other to stay constant
    d. The data contains errors
24. A scatter plot shows all points lying exactly on a straight line with a negative slope. The correlation coefficient is:
    a. r = 0
    b. r = -1
    c. r = 1
    d. Cannot be determined without calculations