

CSC148 Worksheet 5 Solution

Hyungmo Gu

April 17, 2020

Question 1

```
1  """
2  from datetime import date, timedelta
3
4  ...
5
6  misbehaved_1 = Tweet('', date.today(), 'test')
7  misbehaved_1.like(-1)
8
9  yesterday = date.today() - timedelta(days=1)
10 misbehaved_2 = Tweet('john', yesterday, 'test')
11
12 misbehaved_3 = Tweet('john', date.today(), '')
13 """
```

Correct Solution:

```
1  """
2  from datetime import date, timedelta
3
4  ...
5
6  misbehaved_1 = Tweet('', date.today(), 'test')
7  misbehaved_1.like(-1)
8
9  yesterday = date.today() - timedelta(days=1)
10 misbehaved_2 = Tweet('john', yesterday, 'test')
11
12 misbehaved_3 = Tweet('john', date.today(), '')
13
14 misbehaved_4 = Tweet('john doe', date.today(), 'test') # More
15 solution!!
16 """
```

Question 2

- A tweet must receive positive number of likes
- A tweet must have user id without spaces

Correct Solution:

- A tweet must receive **non-negative** number of likes
- **A tweet user id must not be empty.**
- A tweet must have user id without spaces
- **A tweet must have content length less than or equal to 280 characters**

Question 3

- Fix for the property 'a tweet must receive positive number of likes'

```
1  class Tweet:
2      ...
3
4      def like(self, n: int) -> None:
5          """Record the fact that this tweet received <n> likes.
6              These likes are in addition to the ones <self> already has
7              .
8              """
9
10         if n < 0:
11             raise ValueError('n must be non-negative value')
12
13         self.likes += n
```

Correct Solution:

```
1      class Tweet:
2          ...
3
4          def like(self, n: int) -> None:
5              """Record the fact that this tweet received <n>
6              likes.
7              These likes are in addition to the ones <self>
8              already has.
9
10              Precondition: n >= 0 # Correct Solution
```

```

9         """
10
11         self.likes += n
12

```

- Fix for the property ‘a tweet user id must not be empty.’

```

1  class Tweet:
2      ...
3
4      def __init__(self, who: str, when: date, what: str) -> None:
5          """Initialize a new Tweet.
6          """
7
8          if who.strip() == '':
9              raise ValueError("variable 'who' must not be empty")
10
11          self.userid = who
12          self.content = what
13          self.created_at = when
14          self.likes = 0
15

```

Correct Solution:

```

1  class Tweet:
2      ...
3
4      def __init__(self, who: str, when: date, what: str) ->
None:
5          """Initialize a new Tweet.
6
7          Precondition: who.strip() != '' # Correct Solution
8          """
9
10         self.userid = who
11         self.content = what
12         self.created_at = when
13         self.likes = 0
14

```

- Fix for the property ‘a tweet must have user id without spaces’

```

1  class Tweet:
2      ...
3
4      def __init__(self, who: str, when: date, what: str) -> None:
5          """Initialize a new Tweet.
6          """
7
8          if ' ' in who:

```

```

9         raise ValueError("variable 'who' must not have spaces"
10    )
11
12    self.userid = who
13    self.content = what
14    self.created_at = when
15    self.likes = 0

```

Correct Solution:

```

1    class Tweet:
2        ...
3
4    def __init__(self, who: str, when: date, what: str) ->
None:
5        """Initialize a new Tweet.
6
7        Precondition: ' ' not in who # Correct Solution
8        """
9
10       self.userid = who
11       self.content = what
12       self.created_at = when
13       self.likes = 0
14

```

- Fix for the property 'a tweet must have content less than or equal to 280 characters'

```

1    class Tweet:
2        ...
3
4    def __init__(self, who: str, when: date, what: str) -> None:
5        """Initialize a new Tweet.
6        """
7
8        ...
9
10       if len(what) > 280:
11           raise ValueError("variable 'what' must be 280
characters or less")
12
13       self.userid = who
14       self.content = what
15       self.created_at = when
16       self.likes = 0
17

```

Correct Solution:

```
1      class Tweet:
2          ...
3
4      def __init__(self, who: str, when: date, what: str) ->
None:
5          """Initialize a new Tweet.
6
7          Precondition: len(what) <= 280 # Correct Solution
8          """
9
10         self.userid = who
11         self.content = what
12         self.created_at = when
13         self.likes = 0
14
```

Notes:

- Learned that ‘Precondition: len(what) <= 280’ is called **representational invariant**.
- Learned that **representational invariant** is a property of instance attribute that every class must satisfy.

Question 4

a. It is insufficient to implement the *best_percentage*. Consider the following example

1. One or more elements in <teams>with empty strings (i.e. ['', '', ''])
2. One or more elements in <teams>with same name (i.e. ['a', 'a', 'b'])
3. <team1>or <team2>as empty string (i.e. team1 = '', team2 = '')
4. <team1>or <team2>containing the same value (i.e. team1 = 'a', team2 = 'a')
5. <score1>or <score2>as negative value (i.e. score1 = -1, score2 = -4)

b.