

Comments:

REMINDER: If your two required file submissions do not run on the Olin 326 machine on IDLE, the grader will not grade your work. You can use a token to resubmit your project one week after the projects are graded. Otherwise, you will earn one point for each of the following accomplishments.

1. Your files are called `comment.py` and `status.py`. At the top of each file, please include your name, project description, and credit.
 - **Write down the names of the people that you discussed this project with. Even if you helped a classmate during your discussions (as opposed to receiving help), you must write down their name/s.**
 - Also write down the textbook/other Python resources which you used (for example, the Python doc or the address of an online forum such as <http://stackoverflow.com>). *Deduct half a point for a minor mistake.*

2. ____ The contract for your constructor method of your `Comment` class is written accurately, as per the instruction. You have all two instance variables necessary for the `Comment` class. Your instance variables are initialized correctly. Their names are descriptive. *Deduct half a point for a minor mistake.*

Comment:

3. ____ The contract for your constructor method of your `Status` class is written accurately, as per the instruction. You have all four instance variables necessary for the `Status` class. Your instance variables are initialized correctly. Their names are descriptive. *Deduct half a point for a minor mistake.*

Comment:

4. ____ You have correctly written the contract and implementation for all necessary accessor and mutator methods for all the instance variables in the `Comment` class. All these methods have descriptive names (picked by you). *Deduct half a point for a minor mistake.*

Comment:

5. ____ You have correctly written the contract and implementation for all necessary accessor and mutator methods for all the instance variables in the `Status` class. All these methods have descriptive names (picked by you). *Deduct half a point for a minor mistake.*

Comment:

6. ____ The `__str__` method for the `Comment` class and the `__str__` method for the `Status` class have the correct contracts. *Deduct half a point for a minor mistake.*

Comment:

7. ____ The `toggleLike` and `getLikes` methods for the `Status` class have the correct contracts. You also call them exactly `toggleLike` and `getLikes`. *Deduct half a point for a minor mistake.*

Comment:

8. ____ The `addComment` method for the `Status` class has the correct contract, as per the instruction. You also call it exactly `addComment`. *Deduct half a point for a minor mistake.*

Comment:

9. ____ Your class is called `Comment`. Your constructor method for the `Comment` class is implemented as per the instruction. In particular, no error appears when a user tries to create a new `Comment` object, as shown in the examples. **No partial credit.**

Comment:

10. ____ The `__str__` method in the `Comment` works perfectly for all our input strings. **No partial credit.**

Comment:

11. ____ The `__len__` method in the `Comment` is implemented correctly. **No partial credit.**

Comment:

12. ____ The `__contains__` method in the `Comment` is implemented correctly. **No partial credit.**

Comment:

13. ____ The `__getitem__` method in the `Comment` is implemented correctly. **No partial credit.**

Comment:

14. ____ Your class is called `Status`. Your constructor method for the `Status` class is implemented as per the instruction. In particular, no error appears when a user tries to create a new `Status` object, as shown in the examples. **No partial credit.**

Comment:

15. ____ The `__str__` method in the `Status` class is implemented correctly. **No partial credit.**

Comment:

16. ____ The previous bullet point is worth two points. Put the partial credit here in case there is a minor mistake in the implementation of the `__str__` method for the `Status` class. *Deduct half a point for a minor mistake.*

Comment:

17. ____ The `toggleLike` method for the `Status` class works perfectly whenever the given input name is not an existing liker. **No partial credit.**

Comment:

18. ____ The `toggleLike` method for the `Status` class works perfectly whenever the given input name is an existing liker. **No partial credit.**

Comment:

19. ____ The `getLikes` method for the `Status` class returns the correct output for all our test examples. **No partial credit.**

Comment:

20. ____ The `addComment` method for the `Status` class is implemented perfectly. **No partial credit.**

Comment:

_____ Total points