

# JS Advanced Exam

## Problem 2. Story

```
class Story {  
    // TODO: implement this class...  
}
```

### Your Task

Write an Story Class, which supports the described functionality below.

### Functionality

#### constructor(title, creator)

Should have these 4 properties:

- **title** - string
- **creator** - string
- **comments** - private property (empty array by default)
- **likes** - private property (empty array by default)

Note: Choose the most helpful structure for the likes array for you.

#### Getter likes ()

This **function** should print the likes in the following format:

- If there are no likes, the following message should be returned:  
**"{title} has 0 likes"**

- If there is one like, the following message should be returned:  
**"{username} likes this story!"**

- Otherwise, the following message should be **returned**:

**"{username of the first person that liked the story} and {likes - 1} others like this story!"**

#### like (username)

This **function** should increase the likes of the story.

- If the username, has already liked the story, an error with the following message should be thrown:

**"You can't like the same story twice!"**

- If this user is the creator of the story, an error with the following message should be thrown:

**"You can't like your own story!"**

- Otherwise, the following message should be **returned**:

**"{username} liked {title}!"**

## dislike (username)

This **function** should **decrease** the likes of the story.

- If the username, didn't like the story in the first place, an error with the following message should be thrown:

**"You can't dislike this story!"**

- Otherwise, the following message should be **returned**:

**"{username} disliked {title}"**

## comment (username, content, id)

This function should make a new comment or reply to a comment with a given id.

- If the given id is equal to undefined or there is not a comment with that id, you should make a new comment and add it to the comments array. Every comment should have an id (1, 2, 3 ...) which represents the order the comments are submitted. If the comment is made successfully, the following message should be returned:

**"{username} commented on {title}"**

- If there is a comment with the given id, you should add a reply to it. The reply should have an id (1.1, 1.2 ...) constructed from the id of the comment and the order in which the replies are submitted. If the reply is made successfully the following message should be returned:

**"You replied successfully"**

- Comments should have the following structure:

**{Id, Username, Content, Replies}**

- Replies should have the following structure:

**{Id, Username, Content}**

## toString(sortingType)

This function should return the story information in the following format:

- If the sorting type is '**asc**' - The comments and replies should be sorted by **id** in ascending order
- If the sorting type is '**desc**' - The comments and replies should be sorted by **id** in descending order
- If the sorting type is '**username**' - The comments and replies should be sorted by username in ascending order

**"Title: {title}"**

**Creator: {creator}"**

**Likes: {likes}"**

**Comments:**

**-- {id}. {username}: {content}"**

**-- {id}. {username}: {content}"**

**--- {replyId}. {username}: {content}"**

**--- {replyId}. {username}: {content}"**

```
-- {id}. {username}: {content}
..."
```

If there are **no comments** print only the text "Comments:" in the comments section of the **toString** method.

**Note:** For more information see the examples below!

## Submission

Submit only your **Story** class.

## Examples

This is an example how the code is **intended to be used**:

Sample code usage
<pre>Let art = new Story("My Story", "Anny"); art.like("John"); console.log(art.likes); art.dislike("John"); console.log(art.likes); art.comment("Sammy", "Some Content"); console.log(art.comment("Ammy", "New Content")); art.comment("Zane", "Reply", 1); art.comment("Jessy", "Nice :)"); console.log(art.comment("SAMmy", "Reply@", 1)); console.log() console.log(art.toString('username')); console.log() art.like("Zane"); console.log(art.toString('desc'));</pre>
Corresponding output
<pre>John likes this story! My Story has 0 likes Ammy commented on My Story You replied successfully  Title: My Story Creator: Anny Likes: 0 Comments: -- 2. Ammy: New Content -- 3. Jessy: Nice :) -- 1. Sammy: Some Content</pre>

--- 1.2. SAmmy: Reply@

--- 1.1. Zane: Reply

Title: My Story

Creator: Anny

Likes: 1

Comments:

-- 3. Jessy: Nice :)

-- 2. Ammy: New Content

-- 1. Sammy: Some Content

--- 1.2. SAmmy: Reply@

--- 1.1. Zane: Reply