 **Python Programming**

Unit 3 Lab: Object-Oriented Programming

Overview

Welcome to the Unit 3 lab!

Our goal is that at the end of the Unit 5 lab, you'll have an app that prints out the Rotten Tomatoes rating for any movie a user enters. We're getting closer!

Right now, let's use object-oriented programming concepts to improve our code. Specifically, we'll be using dictionaries and classes.

Deliverables

You're going to continue building this locally from where you left off with the last lab. You'll write all of your code in the same `movie_app.py` file.

Run the file from the command line to check your work.

Reminder: On your laptop, you can run the file from your command line with the following:

```
python movie_app.py
```

> **Hint:** Make sure you are printing something out with the `print` statement. Otherwise, you won't see any output from running your program!

Requirements:

1. You have a `Movie` class.
2. Have docstrings on each function.

Directions

Augment the code you wrote for the Unit 2 lab.

Part 1: Docstrings

1. First, for each of your existing functions, add docstrings to document what each function does. (It doesn't hurt to add them to all functions, including `main()`!).

1. Remember that a `docstring` is made with `"""`. For example, your `main()` could look like this: `def main(): """Main is the entry point into the program, and it calls into the search or ratings functions depending on what the user decides to do. """`
2. As you go through this lab, update your docstrings and create new ones for new functions. It will help you, and others in the future, keep track of what each function does.
3. For best practices, follow the Google Docstrings Style Guide!

Part 2: Adding a Class

Part 2a: The Class

1. OK, let's get going! Let's create a class. We're going to have several movies, each of which will have a title and a rating. We can use a `Movie` class as a scaffold to create many movie objects. Near the top of your file, create a class, `Movie`, that takes an argument of `object`. Your `Movie` class should have three functions:
    1. `__init__`, which will take in `self` and `movie_data` (this will be a dictionary containing the title and rating of each movie). The `__init__` function will set two class variables based on the contents of the dictionary: `title` and `rating`.
    2. `get_title()`, a getter function that returns the value of the `title` variable.
    3. `get_rating()`, a getter function that returns the value of the `rating` variable.
2. Now that we have a class, let's set up some functions. Write these *under* the `Movie` class definition, outside of the class scope. First, let's make a function that creates `Movie` objects.
    1. Create a function called `build_movie()` that takes two arguments, `movie_title` and `movie_rating`.
    2. Have it create and return a `Movie` object with those values.

Part 2b: The Class Objects

Now we can make `Movie` objects. Let's look at all the places we're currently using regular movie titles. Can we replace some of them with `Movie` objects?

1. First, let's convert `print_movie_rating()` to consume a `Movie` object. Be sure to reference the class attributes appropriately in the function.
2. Next, convert the list of `movie_titles` to be instead a list of `Movie` objects.
3. Are there any other places we can use the `Movie` object?

Pro tip: If you get confused as to how all these functions interact, it's helpful to draw it out on a piece of paper. Check back to the Requirements section above for an overview.

Phew. You're done! Awesome job.