JavaScript Objects Exercise: Movie Rating App

Exercise 1:

Create an object named `movie` that represents details of a movie. The object should contain the properties `title` (string), `director` (string), `releaseYear` (number), `genres` (array of strings), and `ratings` (array of numbers). Also, add a method `getAverageRating` that returns the average of all ratings.

Exercise 2:

- Add a new property `cast` to the `movie` object created in Exercise 1 without modifying the initial object declaration. The `cast` should be an array of objects, each object representing an actor with properties `name` and `role`.
- Log the second genre in the `genres` array using both dot notation and bracket notation.
- Log the name of the second actor in the `cast` array using both dot notation and bracket notation.

Exercise 3:

- Update the `releaseYear` of the `movie` object to a different year.
- Add a new rating to the `ratings` array.
- Delete the `director` property from the `movie` object.
- Modify the role of the first actor in the `cast` array using both dot and bracket notations.
- Log to if the `movie` object contains the `ratings` property or not.

Exercise 4:

You have bellow an array of movie objects. Each movie object contains a nested object `details` with properties `duration` (number) and `rating` (string e.g., "PG-13").

- Log the `duration` of the first movie in the array.

- Write a function to find the average duration of all movies in the array.

```
  const movies = [
    { title: "Inception", details: { duration: 148,
rating: "PG-13" }},
    { title: "Interstellar", details: { duration: 169,
rating: "PG-13" }}
  ];
```

Exercise 5:

- Merge two movie objects into one.
- Write a code snippet that prevents further changes to a movie object.
- Write a code snippet that prevents new properties from being added to a movie object, but values of existing properties can still be changed.
- Log if a movie object is sealed.
- Log if a movie object is frozen.

Exercise 6:

Given the object `movieDetails` below, write a function that logs all property names and their values separately. Use `Object.keys()` in your solution.

```
const movieDetails = {
    title: "Inception",
    director: "Christopher Nolan",
    releaseYear: 2010,
    ratings: [8, 9, 9.5, 8.5]
};
```

Exercise 7:

Assume you have the `movieCollection` object below where each property is a movie title and its value is the movie's rating. Write a function to increase the rating of a specific movie by 1, but only if the movie's current rating is less than 7. Use `Object.entries()` to find the movie and modify its rating.

```
const movieCollection = {
    "Inception": 9,
    "Interstellar": 8.5,
    "The Dark Knight": 9.5,
    "Prestige": 8
};
```

Exercise 8:

Given the object `movies` below where each key is a movie ID and each value is an object containing `title`, `year`, and `rating`, write a function that returns an array of movies released after the year 2000. Each array element should be an object with all original properties plus an additional `isNewer` property set to true. Utilize `Object.entries()` in your solution.

```
const movies = {
    1: { title: "Inception", year: 2010, rating: 9 },
    2: { title: "The Matrix", year: 1999, rating: 8.5 },
    3: { title: "Interstellar", year: 2014, rating: 8.6 }
};
```

Exercise 9:

You have the `genreRatings` object below that contains movie genres as keys and arrays of numbers (ratings) as values. Write a function that calculates the average rating for each genre and logs a summary. The summary should include the genre name and its average rating. Implement your solution using `Object.keys()`.

```
const genreRatings = {
    action: [8, 9, 7, 10, 8.5],
    sciFi: [8.5, 8, 9, 9.5, 7.5],
    drama: [7, 7.5, 8, 8.5, 9]
};
```

Exercise 10:

Given the object `movieAwards` below where keys are movie titles and values are the awards they won, write a function that creates a new object where the keys are the awards and the values are arrays of movies that won that award. Use `Object.entries()` to traverse the original object and construct the new one.

```
const movieAwards = {
    "Inception": "Best Visual Effects",
    "Interstellar": "Best Visual Effects",
    "The Dark Knight": "Best Supporting Actor",
    "Prestige": "Best Cinematography"
};
```