ES6 Features - Spread, Destructure, Rest

1. Extract Movie Information

Given an object representing a movie, use object destructuring to extract and log the title, year, and rating.

```
const movie = {
  title: "Inception",
  year: 2010,
  rating: 8.8,
  genre: "Science Fiction"
};
```

2. Swap Top Two Movies

Given an array of movie titles, use destructuring to swap the positions of the first two movies in the array.

```
let topMovies = ["The Shawshank Redemption", "The
Godfather", "The Dark Knight", "12 Angry Men"];
```

3. Update Movie Ratings

Given an object where keys are movie titles and values are ratings, write a function that takes this object and an updates object. The function should apply the updates and return a new object using the spread operator.

```
const movieRatings = {
  "The Matrix": 8.7,
  "Inception": 8.8
};

const updates = {
  "Inception": 9.0,
  "The Matrix": 9.1
};
```

4. Combine Movie Lists

You have two arrays of movies collected from two different sources. Combine these arrays into one, removing duplicates, using the spread operator.

```
const list1 = ["The Shawshank Redemption", "The Godfather", "The Dark Knight"];
const list2 = ["Forrest Gump", "Inception", "The Godfather"];
```

5. Extracting Values from Nested Objects

Given a nested object containing movie information including a nested object of cast members, use object destructuring to extract the names of the first two cast members only.

```
const movieDetails = {
  title: "Inception",
  cast: {
    first: "Leonardo DiCaprio",
    second: "Joseph Gordon-Levitt",
    third: "Ellen Page"
  }
};
```

6. Merge Movie Objects

You have two objects representing details about the same movie: one containing the base info and the other containing additional info. Merge these objects using the spread operator.

```
const baseInfo = {
  title: "Inception",
  director: "Christopher Nolan"
};

const additionalInfo = {
  year: 2010,
  rating: 8.8
};
```

7. Expand Movie Genres

Given an object representing a movie's details, including a genres array, write a function that adds new genres to the movie's genres list. The function should be capable of taking any number of new genres and adding them to the existing list without creating duplicates. Use the spread operator to handle the genres array appropriately.

```
const movieDetails = {
  title: "Inception",
  year: 2010,
  rating: 9.0,
  genres: ["Action", "Adventure", "Sci-Fi"]
};
```

8. Flatten Movie Watchlist

Imagine you have a watchlist where each entry is an array representing movies that a user plans to watch per month. Write a function that flattens this watchlist into a single array of movie titles, removing any duplicates. Use the spread operator to concatenate arrays and ensure the watchlist is a flat array of unique titles.

```
const watchlist = [
  ["The Matrix", "Inception"],
  ["Inception", "The Dark Knight"],
  ["The Dark Knight", "Interstellar"]
];
```

9. Movie Rating Updater

Given an array of movies where each movie is an object containing the title and rating, write a function that increases the rating of a specific movie by a given amount. Use the `map` method and object destructuring in the function to simplify your code.

```
const movies = [
  { title: "The Matrix", rating: 8.7 },
  { title: "Inception", rating: 8.8 }
];
```

10. Cast List Organizer

Given an object representing a movie, which includes the title, director, and an array of cast members, write a function that returns a new object. The new object should include the movie's title and an array of cast members' first names only. Use the `map` array method and destructuring to extract these details.

```
const movieDetails = {
  title: "The Dark Knight",
  director: "Christopher Nolan",
  cast: ["Christian Bale", "Heath Ledger", "Aaron
Eckhart"]
};
```