Array of Movies

```
let movies = [
  {
    id: 1,
    title: "The Shawshank Redemption",
    genre: "Drama",
    director: "Frank Darabont",
    releaseYear: 1994,
    rating: 9.3,
    votes: 2345890,
  },
  {
    id: 2,
    title: "The Godfather",
    genre: "Crime, Drama",
    director: "Francis Ford Coppola",
    releaseYear: 1972,
    rating: 9.2,
    votes: 1620360,
  },
  {
    id: 3,
    title: "The Godfather: Part II",
    genre: "Crime, Drama",
    director: "Francis Ford Coppola",
    releaseYear: 1974,
    rating: 9.0,
    votes: 1133175,
  },
  {
    id: 4,
    title: "The Dark Knight",
    genre: "Action, Crime, Drama",
    director: "Christopher Nolan",
    releaseYear: 2008,
    rating: 9.0,
    votes: 2303232,
  },
  {
    id: 5,
    title: "12 Angry Men",
    genre: "Crime, Drama",
    director: "Sidney Lumet",
    releaseYear: 1957,
    rating: 9.0,
    votes: 689845,
```

```
  },
];
```

Strings

1. **Movie Initials:** Create a function, `getMovieInitials(title)`, that takes a movie title and returns its initials.
    - Input: "The Dark Knight"
    - Output: "TDK"
2. **Genre List:** Implement a function, `splitGenres(genres)`, that takes a string of genres separated by commas and converts it into an array of individual genres.
    - Input: "Action, Adventure, Sci-Fi"
    - Output: ["Action", "Adventure", "Sci-Fi"]

Arrow Functions

3. **Arrow Functions with Multiple Operations:** Given a function `getTotalMovieLength` that calculates the total length of all movies and returns a string, convert it into an arrow function.

```
function getTotalMovieLength(movies) {
  let totalLength = 0;
  for (let i = 0; i < movies.length; i++) {
    totalLength += movies[i].length;
  }
  return 'Total movie length is ' + totalLength + ' minutes';
```

Array Methods

1. **Display Movie Titles:** Given an array of movie objects, return a new array that contains only the titles of each movie.
    - Example: Given the input data from above, the function should return: ["The Shawshank Redemption", "The Godfather", "The Godfather: Part II", "The Dark Knight", "12 Angry Men"]
2. **Highly Rated Movies:** Given an array of movie objects and a minimum rating value, return a new array containing only movies with a rating equal to or greater than the provided value.
    - Example: Given the input data and a rating of 9.1, the function should return an array containing only the objects for "The Shawshank Redemption" and "The Godfather."
3. **Average Movie Rating:** Calculate and return the average movie rating from an array of movie objects.
    - Example: Given the input data, the function should return approximately 9.1.
4. **Sort Movies by Votes:** Given an array of movie objects, return a new array sorted by the number of votes in descending order.

- Example: Given the input data, the first movie in the returned array should be "The Shawshank Redemption," and the last should be "12 Angry Men."

Objects

Exercise 1: Creating the Movie Database

1. Define an object named `movieDatabase` that will store our movies and their ratings.
2. The `movieDatabase` object should have a `movies` property that is initialized as an empty array. Each movie will be an object stored in this array, with properties: title (a string), ratings (an array of numbers), and averageRating (a number).
3. Add a method named `addMovie` to the `movieDatabase` object. This method should take a movie title as a parameter and create a new movie object with the given title, an empty ratings array, and an averageRating of 0. The new movie should then be added to the `movies` array.
4. Add a method named `removeMovie` to `movieDatabase`. This method should take a movie title as a parameter and remove the corresponding movie object from the `movies` array.

Exercise 2: Adding and Removing Ratings

1. Add a method named `addRating` to `movieDatabase`. This method should take a movie title and a rating as parameters. It should find the movie with the given title in the `movies` array and add the rating to its ratings array.
2. Add a method named `removeRating` to `movieDatabase`. This method should take a movie title and a rating as parameters. It should find the movie with the given title in the `movies` array and remove the given rating from its ratings array.

Exercise 3: Searching Movies

1. Add a method named `searchMovie` to `movieDatabase`. This method should take a string as a parameter and return an array of all movies whose title contains that string (ignoring case).

Note: The solutions are not provided here. These are exercises to practice JavaScript programming.