# Beginner Java Exercises Focused on ArrayList, Comparable, equals, toString, and Collections.sort

## Exercise 1: Creating a Simple Class and Using toString()

Objective: Understand how to create a class with attributes and override the toString() method.  
  
Instructions:  
1. Create a class called Movie with the following attributes:  
 - title (String)  
 - director (String)  
 - releaseYear (int)  
2. Create a constructor that initializes the above attributes.  
3. Override the toString() method to provide a formatted string representation of a Movie object, displaying the title, director, and release year.  
4. In the main method, create an ArrayList of Movie objects and add at least three movies to the list.  
5. Use a loop to print each movie in the list, observing the output from the toString() method.  
  
Expected Output:  
The output should look like this:  
```java  
Movie{title='Inception', director='Christopher Nolan', releaseYear=2010}  
Movie{title='Parasite', director='Bong Joon-ho', releaseYear=2019}  
Movie{title='The Matrix', director='Lana Wachowski, Lilly Wachowski', releaseYear=1999}  
```

## Exercise 2: Implementing equals() and Testing Equality

Objective: Learn to override the equals() method to compare objects based on specific attributes.  
  
Instructions:  
1. Reuse the Movie class from Exercise 1.  
2. Override the equals() method to consider two Movie objects equal if they have the same title and releaseYear. The director field should not affect equality.  
3. In the main method, create two Movie objects with the same title and releaseYear but different directors.  
4. Use the equals() method to check if these two movies are considered equal.  
5. Print a message stating whether the two movies are equal or not.  
  
Expected Output:  
If you compare two movies with the same title and year, the output should indicate that they are equal, like:  
```java  
Are the movies equal? true  
```

## Exercise 3: Sorting with Comparable and Collections.sort()

Objective: Learn to implement the Comparable interface and use Collections.sort() to sort objects in an ArrayList.  
  
Instructions:  
1. Modify the Movie class to implement the Comparable interface.  
2. Override the compareTo() method to sort movies by releaseYear in ascending order.  
3. In the main method, create an ArrayList of Movie objects and add at least five movies with different release years.  
4. Use Collections.sort() to sort the list of movies by release year.  
5. Print the sorted list to verify the order.  
  
Expected Output:  
After sorting, movies should appear in the order of their release years, like:  
```java  
Movie{title='The Matrix', director='Lana Wachowski, Lilly Wachowski', releaseYear=1999}  
Movie{title='Inception', director='Christopher Nolan', releaseYear=2010}  
Movie{title='Parasite', director='Bong Joon-ho', releaseYear=2019}  
Movie{title='Dune', director='Denis Villeneuve', releaseYear=2021}  
```

## Summary of Exercise Goals

- Exercise 1: Introduces ArrayList, class creation, and toString() method for readable output.  
- Exercise 2: Builds on equals() method for custom object comparison based on specific attributes.  
- Exercise 3: Introduces Comparable and Collections.sort() to sort custom objects by an attribute (releaseYear).