# Beginner Java Exercises

## Exercise 1: Introduction to Classes and Basic Printing

Objective: Understand how to create a class with attributes, and learn how to print an object in a readable format.  
  
Instructions:  
1. Create a new class called `Person` with the following attributes:  
 - `name` (String)  
 - `age` (int)  
 - `city` (String)  
2. Create a constructor for the `Person` class that initializes these attributes.  
3. Override the `toString()` method to return a readable description of the `Person`.  
4. In the main method, create a new `Person` object with a name, age, and city.  
5. Print the `Person` object to see the result.  
  
Expected Output:  
When you print the `Person` object, it should look something like:  
```java  
Person{name='Alice', age=30, city='New York'}  
```

## Exercise 2: Working with an ArrayList

Objective: Learn to use `ArrayList` for managing a list of objects and practice basic list operations.  
  
Instructions:  
1. Reuse the `Person` class from Exercise 1.  
2. In the main method, create an `ArrayList` of `Person` objects and add at least three people to the list.  
3. Write a loop to print each person in the list.  
4. Add code to remove one person from the list, and print the list again to verify the change.  
  
Example:  
- Create an ArrayList of people with names "Alice", "Bob", and "Charlie".  
- Remove "Charlie" from the list and print the updated list.  
  
Expected Output:  
Before removing "Charlie":  
```java  
Person{name='Alice', age=30, city='New York'}  
Person{name='Bob', age=25, city='Chicago'}  
Person{name='Charlie', age=35, city='Los Angeles'}  
```  
After removing "Charlie":  
```java  
Person{name='Alice', age=30, city='New York'}  
Person{name='Bob', age=25, city='Chicago'}  
```

## Exercise 3: Implementing `equals` and Sorting with `Collections.sort()`

Objective: Learn to override the `equals()` method for comparing objects and practice sorting a list of objects.  
  
Instructions:  
1. Modify the `Person` class to:  
 - Override the `equals` method to consider two `Person` objects equal if they have the same `name` and `age`.  
2. In the main method:  
 - Create an `ArrayList` of `Person` objects with at least four people.  
 - Test the `equals` method by checking if two people with the same name and age are considered equal.  
3. Implement the `Comparable` interface for `Person` to sort by `name` in alphabetical order.  
4. Use `Collections.sort()` to sort the list of people by `name` and print the sorted list.  
  
Example:  
- Test the `equals()` method with two `Person` objects with the same name and age, e.g., "Alice" and 30.  
- Sort and print the list of people by their names.  
  
Expected Output:  
After sorting:  
```java  
Person{name='Alice', age=30, city='New York'}  
Person{name='Bob', age=25, city='Chicago'}  
Person{name='Charlie', age=35, city='Los Angeles'}  
Person{name='David', age=28, city='Boston'}  
```