



Comparator Interface

Implement different sort orders with the Comparator interface's `compare()` method, and pass the Comparator as a parameter to `List.sort()`. Your implementation must explicitly handle null inputs in the `compare()` method to avoid a null pointer exception.

Namespace

System

- [Comparator Methods](#)
- [Comparator Example Implementation](#)

Use the Comparator interface to impose different kinds of sorting.

Comparator Methods

The following are methods for `Comparator`.

- [`compare\(var1, var2\)`](#)
Compares the two arguments and returns a negative integer, zero, or a positive integer depending on whether the first argument is less than, equal to, or greater than the second argument.

`compare(var1, var2)`

Compares the two arguments and returns a negative integer, zero, or a positive integer depending on whether the first argument is less than, equal to, or greater than the second argument.

Signature

```
public Integer compare(T var1, T var2)
```

Parameters

var1

Type: `T`

`T` - The type determined by the parameterized type of the Comparator. For example, if the class implements `Comparator<Account>` then *var1* and *var2* are of type `Account`.

var2

Type: `T`

`T` - The type determined by the parameterized type of the Comparator. For example, if the class implements `Comparator<Account>` then *var1* and *var2* are of type `Account`.

Return Value

Type: [Integer](#)

Comparator Example Implementation

Use the Comparator interface to impose different kinds of sorting.



```

private Long id;
private String name;
private Integer yearJoined;

// Constructor
public Employee(Long i, String n, Integer y) {
    id = i;
    name = n;
    yearJoined = y;
}

public String getName() { return name; }
public Integer getYear() { return yearJoined; }
}

```

```

// Class to compare Employees by name
public class NameCompare implements Comparator<Employee> {
    public Integer compare(Employee e1, Employee e2) {
        if(e1?.getName() == null && e2?.getName() == null) {
            return 0;
        } else if(e1?.getName() == null) {
            return -1;
        } else if(e2?.getName() == null) {
            return 1;
        }
        return e1.getName().compareTo(e2.getName());
    }
}

// Class to compare Employees by year joined
public class YearCompare implements Comparator<Employee> {
    public Integer compare(Employee e1, Employee e2) {
        // Guard against null operands for '<' or '>' operators because
        // they will always return false and produce inconsistent sorting
        Integer result;
        if(e1?.getYear() == null && e2?.getYear() == null) {
            result = 0;
        } else if(e1?.getYear() == null) {
            result = -1;
        } else if(e2?.getYear() == null) {
            result = 1;
        } else if (e1.getYear() < e2.getYear()) {
            result = -1;
        } else if (e1.getYear() > e2.getYear()) {
            result = 1;
        } else {
            result = 0;
        }
        return result;
    }
}

```

The following example tests the implementation:

```

@isTest
private class EmployeeSortingTest {
    @isTest
    static void sortWithComparators() {
        List<Employee> empList = new List<Employee>();
        empList.add(new Employee(101, 'Joe Smith', 2020));
        empList.add(new Employee(102, 'J. Smith', 2020));
        empList.add(new Employee(25, 'Caragh Smith', 2021));
        empList.add(new Employee(105, 'Mario Ruiz', 2019));
        // Sort by name
        NameCompare nameCompare = new NameCompare();
        empList.sort(nameCompare);
    }
}

```



```
// Expected order: Mario Ruiz, J. Smith, Joe Smith, Caragh Smith
Assert.areEqual('Mario Ruiz', emplist.get(0).getName());
}
}
```

DID THIS ARTICLE SOLVE YOUR ISSUE?
Let us know so we can improve!

Share your feedback



DEVELOPER CENTERS

- Heroku
- MuleSoft
- Tableau
- Commerce Cloud
- Lightning Design System
- Einstein
- Quip

POPULAR RESOURCES

- Documentation
- Component Library
- APIs
- Trailhead
- Sample Apps
- Podcasts
- AppExchange

COMMUNITY

- Trailblazer Community
- Events and Calendar
- Partner Community
- Blog
- Salesforce Admins
- Salesforce Architects

© Copyright 2025 Salesforce, Inc. [All rights reserved.](#) Various trademarks held by their respective owners. Salesforce, Inc. Salesforce Tower, 415 Mission Street, 3rd Floor, San Francisco, CA 94105, United States

[Privacy Information](#) [Terms of Service](#) [Legal](#) [Use of Cookies](#) [Trust](#) [Cookie Preferences](#)
☒ [Your Privacy Choices](#) [Responsible Disclosure](#) [Contact](#)