

# COMP 250

Assignment Project Exam Help

## INTRODUC TER SCIENCE

<https://eduassistpro.github.io/>

Week 8-2 : OODs  
Add WeChat edu\_assist\_pro

Giulia Alberini, Fall 2020

# WHAT ARE WE GOING TO DO IN THIS VIDEO?



- Java interface Comparable
- ## Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

# JAVA Comparable INTERFACE

- The Java Comparable interface is used to define an ordering on objects of user-defined class.

Assignment Project Exam Help

<https://eduassistpro.github.io/>

- Why would you want that? Well, if you have a list of objects from a given class you might want to be able to sort them.

Add WeChat edu\_assist\_pro

- Comparable is part of java.lang package and contains only one method named compareTo(Object).

# JAVA Comparable INTERFACE

Assignment Project Exam Help

```
public interface Comparable<T> {  
    int compareTo(T o);  
}
```

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

<https://docs.oracle.com/javase/7/docs/api/java/lang/Comparable.html>

# JAVA Comparable INTERFACE

Some of the methods from certain Java classes use `compareTo()` in their implementation. To function correctly, they assume to be working with Comparable generic types. Examples:

- `sort()` from Arrays.

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

# JAVA Comparable INTERFACE

Some of the methods from certain Java classes use `compareTo()` in their implementation. To function correctly, they assume to be working with Comparable generic types. Examples:

- `sort()` from `Collection` <https://eduassistpro.github.io/>

Add WeChat `edu_assist_pro`

String **IMPLEMENTS** Comparable

Assignment Project Exam Help

~~<https://eduassistpro.github.io/>~~

Add WeChat edu\_assist\_pro

<https://docs.oracle.com/javase/7/docs/api/java/lang/String.html>



## CLASSES THAT IMPLEMENT Comparable

- Character, Integer, Float, Double, BigInteger, etc. all implement Comparable

<https://eduassistpro.github.io/>

- You cannot compare objects of `Comparable` using the "<" operator. Instead use `compareTo()`.

## HOW TO IMPLEMENT Comparable

- **Add** `implements Comparable` in the definition of the class.

Assignment Project Exam Help

- **Implement** `compareTo` <https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

```
public class T implements Comparable<T>{  
    public int compareTo(T o) {...}  
}
```

## REQUIREMENT FOR IMPLEMENTING compareTo()

**Consider two variable `t1` and `t2` or type `T`. Then,**

# Assignment Project Exam Help

```
t1.compareTo(t2) {
    int i, if t1 < t2
    https://eduassistpro.github.io/, if t1 = t2
    Add WeChat edu_assist_pro, if t1 > t2
}
```

The relation should also be anticommutative and transitive.

## Highly recommended:

```
(t1.compareTo(t2)==0) == (t1.equals(t2))
```

## EXAMPLE - CIRCLE

- Sometimes deciding how to compare elements of a given type can be straightforward.

- Let's think about the data

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

```
public class Circle {  
    private double r;  
    :  
}
```

- How should we implement `compareTo()` and `equals()` in order to establish a *natural ordering* between elements of type `Circle`?

## EXAMPLE - CIRCLE

- How should we implement `compareTo()` and `equals()` in order to establish a *natural ordering* between elements of type `Circle`?

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

We could simply compare their radius (or their area).

## CIRCLE – compareTo ()

```
public class Circle extends Shape implements Comparable<Circle>{
    private double radius = 5;

    public int compareTo(Circle c) {
        if (this.radius < c.radius)
            return -1;
        else if (this.radius > c.radius)
            return 1;
        else
            return 0;
    }

    public boolean equals(Object obj) {
        return obj instanceof Circle && ((Circle) obj).radius == this.radius;
    }
}
```

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

## EXAMPLE – ORC

- Other times, is not so straightforward. Suppose we have created a new data type `Orc`.

- How should we compare `type?`

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro



Base on their name? On their height? On their weapon? On who is scarier?

## ORC – compareTo () TAKE 1

```
public class Orc implements Comparable<Orc> {  
    private String name;  
    private int height;  
    private Weapon w;  
    public int compareTo(Orc o)  
        if(this.height < o.height)  
            return -1;  
        } else if(this.height == o.height)  
            return 0;  
        } else {  
            return 1;  
        }  
    }  
}
```

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

- Note that in this case we probably don't want to consider two Orcs with the same height to be equal. This implies that the implementation of compareTo() violates the Java API recommendations.
- Such violation should be clearly indicated using the following language: "Note: this class has a natural ordering that is inconsistent with equals."



## ORC – compareTo () TAKE 2

```
public class Orc implements Comparable<Orc> {  
    private String name;  
    private Integer height;  
    private Weapon w;  
    public int compareTo(Orc o)  
    {  
        int result = this.w.compareTo(o.w);  
        if(result==0) {  
            result = this.height.compareTo(o.height);  
        }  
        if(result == 0) {  
            result = this.name.compareTo(o.name);  
        }  
        return result;  
    }  
}
```

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

- We can also use `compareTo()` to compare multiple characteristics.
- Generally, it is better to reuse existing code than to write our own. Thus, in this case, we can use the `compareTo()` methods from other classes to.

## TO RECAP

- `Comparable` defines a natural ordering.
- If you define a new `d` that makes sense to you, then you should implement `Comparable` to define a natural ordering on objects of such type.

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

An orange paint roller with a red handle, positioned horizontally. The roller is partially filled with orange paint, and there are orange paint splatters and drips around it. The text "Coming Soon" is written in white on the orange background of the roller.

# Coming Soon

## Assignment Project Exam Help

In the next

- Iterable <https://eduassistpro.github.io/>  
Add WeChat edu\_assist\_pro