

Relational Databases with MySQL Week 11 Assignment

Points possible: 70

Category	Criteria	% of Grade
Functionality	Does the code work?	25
Organization	Is the code clean and organized? Proper use of white space, syntax, and consistency are utilized. Names and comments are concise and clear.	25
Creativity	Student solved the problems presented in the assignment using creativity and out of the box thinking.	25
Completeness	All requirements of the assignment are complete.	25

Instructions: Complete the coding steps. Take screenshots of the code and of the running program (make sure to get screenshots of all required functionality) and paste them in this document where instructed below. Create a new repository on GitHub for this week's assignments and push this document to the repository. Additionally, push the Java project to the same repository. Add the URL for this week's repository to this document where instructed and submit this document to your instructor when complete.

Coding Steps:

1. Create a class of whatever type you want (Animal, Person, Camera, Cheese, etc.).
 - a. Do not implement the Comparable interface.
 - b. Add a name instance variable so that you can tell the objects apart.
 - c. Add getters, setters and/or a constructor as appropriate.
 - d. Add a toString method that returns the name and object type (like "Pentax Camera").
 - e. Create a static method named `compare` in the class that returns an int and takes two of the objects as parameters. Return -1 if parameter 1 is "less than" parameter 2. Return 1 if parameter 1 is "greater than" parameter 2. Return 0 if the two parameters are "equal".
 - f. Create a static list of these objects, adding at least 4 objects to the list.
 - g. In another class, write a method to sort the objects using a Lambda expression using the compare method you created earlier.
 - h. Write a method to sort the objects using a Method Reference to the compare method you created earlier.
 - i. Create a main method to call the sort methods.
 - j. Print the list after sorting (`System.out.println`).

2. Create a new class with a main method. Using the list of objects you created in the prior step.
 - a. Create a Stream from the list of objects.
 - b. Turn the Stream of object to a Stream of String (use the map method for this).
 - c. Sort the Stream in the natural order. (Note: The String class implements the Comparable interface, so you won't have to supply a Comparator to do the sorting.)
 - d. Collect the Stream and return a comma-separated list of names as a single String. Hint: use `Collectors.joining(", ")` for this.
 - e. Print the resulting String.
3. Create a new class with a main method. Create a method (method a) that accepts an Optional of some type of object (Animal, Person, Camera, etc.).
 - a. The method should return the object unwrapped from the Optional if the object is present. For example, if you have an object of type Cheese, your method signature should look something like this:

```
public Cheese cheesyMethod(Optional<Cheese> optionalCheese) {...}
```
 - b. The method should throw a `NoSuchElementException` with a custom message if the object is not present.
 - c. Create another method (method b) that calls method a with an object wrapped by an Optional. Show that the object is returned unwrapped from the Optional (i.e., print the object).
 - d. Method b should also call method a with an empty Optional. Show that a `NoSuchElementException` is thrown by method a by printing the exception message. Hint: catch the `NoSuchElementException` as parameter named "e" and do `System.out.println(e.getMessage())`.
 - e. Note: your method should handle the Optional as shown in the video on Optionals using the `orElseThrow` method. For the missing object, you must use a Lambda expression in `orElseThrow` to return a `NoSuchElementException` with a custom message.

Screenshots of Code:

Color.java U x SortColors.java U StreamColor.java U OptionalColor.java U

src > week11 > model > Color.java > Color

```
1  package week11.model;
2
3
4  public class Color {
5      private String colorName;
6
7      public Color (String name){
8          this.colorName = name;
9      }
10
11     @Override
12     public String toString() {
13         return colorName;
14     }
15
16     public String getColorName() {
17         return colorName;
18     }
19
20     public static int compareColors (Color c1, Color c2){
21
22         if (c1.getColorName().compareTo(c2.getColorName()) > 0){
23             return 1;
24         } else if (c1.getColorName().compareTo(c2.getColorName()) < 0){
25             return -1;
26         }
27         return 0;
28     }
29
30 }
31
```

```
SortColors.java - PromineoWeek11 - Visual Studio Code

... Color.java U SortColors.java U X StreamColor.java U OptionalColor.java U

src > week11 > SortColors.java > SortColors > colors
1 package week11;
2
3 import java.util.ArrayList;
4 import java.util.Comparator;
5 import java.util.List;
6
7 import week11.model.Color;
8
9
10 public class SortColors {
11     private Comparator<Color> comp = null;
12     private List<Color> colors = new ArrayList<>(List.of(new Color(name: "Red"), new Color(name: "Green"), new Color(name: "Blue"),
13     new Color(name: "Cyan"), new Color(name: "Yellow"), new Color(name: "Magenta"))));
14
15     public static void main(String[] args) throws Exception {
16         new SortColors().run();
17     }
18
19     private void run() {
20         System.out.println(x: "Original list of colors");
21         colors.forEach(System.out::println);
22         System.out.println();
23
24         //Lambda
25         System.out.println(x: "Color list sorted using a Lambda expression");
26         List<Color> sortedLambda = new ArrayList<>(colors);
27         sortLambda(sortedLambda);
28         sortedLambda.forEach(color -> System.out.println(color.getColorName()));
29         System.out.println();
30
31         //Method Reference
32     }
```

```
SortColors.java - PromineoWeek11 - Visual Studio Code

Color.java U SortColors.java U X StreamColor.java U OptionalColor.java U

src > week11 > SortColors.java > SortColors > colors
32 //Method Reference
33 System.out.println(x: "Color list sorted using a Method Reference expression");
34 List<Color> sortedRefM = new ArrayList<>(colors);
35 sortMethodRef(sortedRefM);
36 sortedRefM.forEach(System.out::println);
37 System.out.println();
38
39 //original list again
40 System.out.println(x: "Original list of colors again for reference");
41 colors.forEach(System.out::println);
42
43 }
44
45 private void sortLambda(List<Color> sortColors){
46     comp = (c1, c2) -> Color.compareColors(c1, c2);
47     sortColors.sort(comp);
48 }
49
50 private void sortMethodRef(List<Color> sortColors){
51     comp = Color::compareColors;
52     sortColors.sort(comp);
53 }
54
55 }
56 }
```

```
StreamColor.java - PromineoWeek11 - Visual Studio Code
Color.java U  SortColors.java U  StreamColor.java U x  OptionalColor.java U

src > week11 > StreamColor.java > ...
1 package week11;
2
3 import java.util.ArrayList;
4 import java.util.List;
5 import java.util.stream.Collectors;
6
7
8 import week11.model.Color;
9
10 public class StreamColor {
11
12     private List<Color> colors = new ArrayList<>(List.of(new Color(name: "Red"), new Color(name: "Green"), new Color(name: "Blue"),
13         new Color(name: "Cyan"), new Color(name: "Yellow"), new Color(name: "Magenta")));
14
15     Run | Debug
16     public static void main(String[] args) {
17         new StreamColor().run();
18     }
19
20     private void run() {
21
22         String sortedColors = colors.stream()
23             .map(Color::getColorName)
24             .sorted()
25             .collect(Collectors.joining(delimiter: ", "));
26         System.out.println(sortedColors);
27     }
28
29 }
30
```

```
Color.java U  SortColors.java U  StreamColor.java U  OptionalColor.java U x

src > week11 > OptionalColor.java > OptionalColor > main(String[])
1 package week11;
2
3 import java.util.ArrayList;
4 import java.util.List;
5 import java.util.NoSuchElementException;
6 import java.util.Optional;
7
8 import week11.model.Color;
9
10 public class OptionalColor {
11     private List<Color> colors = new ArrayList<>(List.of(new Color(name: "Red"), new Color(name: "Green"), new Color(name: "Blue"),
12         new Color(name: "Cyan"), new Color(name: "Yellow"), new Color(name: "Magenta")));
13     Run | Debug
14     public static void main(String[] args) {
15         new OptionalColor().methodB();
16     }
17
18     private Color colorMethod(Optional<Color> optionalColor){
19         return optionalColor.orElseThrow(() -> new NoSuchElementException(s: "This color isn't in our list!"));
20     }
21
22     private void methodB() {
23         try {
24             System.out.println("The color you selected was: " + colorMethod(Optional.of(colors.get(index: 2))).toString());
25         } catch (Exception e) {
26             System.out.println(e.getMessage());
27         }
28
29         try {
30             System.out.println("The color you selected was: " + colorMethod(Optional.empty()).toString());
31         } catch (Exception e) {
32             System.out.println(e.getMessage());
33         }
34     }
35 }
36
```

Screenshots of Running Application Results:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  GITLENS

in' 'week11.SortColors'
Original list of colors
Red
Green
Blue
Cyan
Yellow
Magenta

Color list sorted using a Lambda expression
Blue
Cyan
Green
Magenta
Red
Yellow

Color list sorted using a Method Reference expression
Blue
Cyan
Green
Magenta
Red
Yellow

Original list of colors again for reference
Red
Green
Blue
Cyan
Yellow
Magenta
PS C:\Users\Merel\Promineo\PromineoWeek11\PromineoWeek11>
```

```
Install the latest PowerShell for new features and improvements! https://aka.ms/PowerShellLatest

PS C:\Users\Merel\Promineo\PromineoWeek11\PromineoWeek11> & 'C:\Program Files\PowerShell\7\powershell.exe' --enable-preview -XX:+ShowCodeDetailsInExceptionMessages -cp'
Blue, Cyan, Green, Magenta, Red, Yellow
PS C:\Users\Merel\Promineo\PromineoWeek11\PromineoWeek11>
```

```
p=transport=dt_socket,server=n,suspend=y,address=localhost:5794
in' 'week11.OptionalColor'
The color you selected was: Blue
This color isn't in our list!
PS C:\Users\Merel\Promineo\PromineoWeek11\PromineoWeek11> 
```

URL to GitHub Repository:

<https://github.com/MerelOhler/PromineoWeek11>