Java Collection Exercise

```
public class Customer {
  private Long id;
  private String name;
  private Integer tier;
  ...
}


public class Order {
  private Long id;
  private LocalDate orderDate;
  private LocalDate deliveryDate;
  private String status;
  //@ManyToOne
  private Customer customer;
  Set<Product> products;
  ..
}


public class Product {
  private Long id;
  private String name;
  private String category;
  private Double price;
  //@ManyToMany(mappedBy = "products")
  private Set<Order> orders;
   ..
}
```

Exercise 1 — Obtain a list of products belongs to category “Books” with price > 100

Exercise 2 — Obtain a list of order with products belong to category “Baby”

Exercise 3 — Obtain a list of product with category = “Toys” and then apply 10% discount

Exercise 4 — Obtain a list of products ordered by customer of tier 2 between 01-Feb-2021 and 01-Apr-2021

Exercise 5 — Get the cheapest products of “Books” category

Exercise 6 — Get the 3 most recent placed order

Exercise 7 — Get a list of orders which were ordered on 15-Mar-2021, log the order records to the console and then return its product list

Exercise 8 — Calculate total lump sum of all orders placed in Feb 2021

Exercise 9 — Calculate order average payment placed on 14-Mar-2021

Exercise 10 — Obtain a collection of statistic figures (i.e. sum, average, max, min, count) for all products of category “Books”

Exercise 11 — Obtain a data map with order id and order’s product count

Exercise 12 — Produce a data map with order records grouped by customer

Exercise 13 — Produce a data map with order record and product total sum

Exercise 14 — Obtain a data map with list of product name by category

Exercise 15 — Get the most expensive product by category

https://blog.devgenius.io/15-practical-exercises-help-you-master-java-stream-api-3f9c86b1cf82