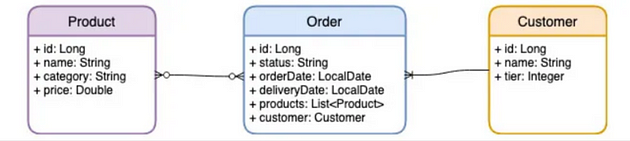
1. Write Java 8 program to multiply 3 with all the elements in the list and print?
2. Print current date and time using Java Date and Time API?
3. Print duplicate element of given integer list ?
4. Print n random numbers using Java 8?
5. Get all list of products that belong to books category and product price >200 using java 8 features?
6. Get cheapest product of Iphone category using java 8?



public class Order{  
  
 long id;  
 String status;  
 LocalDate orderDate;  
 LocalDate deliveryDate;  
 List<Product> products;  
 Customer customer;  
}

public class Product {  
  
 long id;  
 String name;  
 String category;  
 Double price;  
  
 }

public class Customer {  
 long id;  
 String name;  
 int tier;  
}

public class OrderApplication {  
  
 public static void main(String[] args) {  
  
 Scanner sc = new Scanner(System.in);  
 Product productOne = new Product(13, "Bottle", "Book", 120.0);  
 Product productTwo = new Product(14, "Pencil", "Toy", 80.0);  
 Product productThree = new Product(15, "Pen", "Toy", 140.0);  
 Product productfour = new Product(16, "book2", "Book", 190.0);  
  
 Set<Product> productSet = new HashSet<>();  
 productSet.add(productOne);  
 productSet.add(productTwo);  
 productSet.add(productThree);  
  
 List<Product> products = new ArrayList<>();  
 products.add(productOne);  
 products.add(productThree);  
 List<Product> products1 = new ArrayList<>();  
 products1.add(productTwo);  
 products1.add(productfour);  
 Customer customer1 = new Customer(1, "Anuj", 1);  
 Customer customer2 = new Customer(2, "Rekha", 2);  
  
 Set<Order> orderSet = new HashSet<>();  
 orderSet.add(new Order(123, "Delivered", LocalDate.now(), LocalDate.now(), products, customer1));  
 orderSet.add(new Order(131, "Not Delivered", LocalDate.now(), LocalDate.now(), products1, customer2));  
}

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

productSet.stream().filter(product -> product.price > 100.0).forEach(System.out::println);

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

List<Product> productList = productSet.stream().filter(product -> product.getCategory().equals("Toy")).  
 map(product -> new Product(product.id, product.name, product.category, product.price - (product.price / 100) \* 10)).collect(Collectors.toList());

Get the cheapest product of “Books” category

Product minByPrice = productSet  
 .stream()  
 .filter(p->p.getCategory().equals("Book"))  
 .min(Comparator.comparing(Product::getPrice))  
 .orElseThrow(NoSuchElementException::new);

4. Obtain a list of order with products belong to category “Books”

List<Order> orders = orderSet.stream().filter(order -> order.getProducts().stream().  
 anyMatch(p -> p.getCategory().equalsIgnoreCase("Book"))).collect(Collectors.toList());

5. Get a list of orders which were ordered on 20-Feb-2023, log the order records to the console and then return its product list

orderSet.stream().filter(order -> order.orderDate.isEqual(LocalDate.parse("2023-02-20"))).forEach(System.out::println);