Design a class named Product with these data members.product id of type int , product name of type String , category of type String and price of type double.

- generate a setter/getter for each data member and all arg constructor and toString() method.
- Add a method named **withPrice()** which will compare the current object price with a price parameter that sent to the method and return the current object in case of equality.otherwise return new product with the current object id , name , category and the price parameter that sent to the function.
  Hint : the method body like this `public Product withPrice(double price)`
- Use addDummyData method to add dummy data

```
public List<Product> addDummyData()
{
List<Product> products = new ArrayList<>();
Collections.addAll(products,
new Product(1, "omnis quod consequatur", "Games", 184.83),
new Product(2, "vel libero suscipit", "Toys", 12.66),
new Product(3, "non nemo iure", "Grocery", 498.02),
new Product(4, "voluptatem voluptas aspernatur", "Toys", 536.80),
new Product(5, "animi cum rem", "Games", 458.20),
new Product(6, "dolorem porro debitis", "Toys", 146.52),
new Product(7, "aspernatur rerum qui", "Books", 656.42),
new Product(8, "deleniti earum et", "Baby", 41.46),
new Product(9, "voluptas ut quidem", "Books", 697.57),
new Product(10, "eos sed debitis", "Baby", 366.90),
new Product(11, "laudantium sit nihil", "Toys", 95.50),
new Product(12, "ut perferendis corporis", "Grocery", 302.19),
new Product(13, "sint voluptatem ut", "Toys", 295.37),
new Product(14, "quos sunt ipsam", "Grocery", 534.64),
new Product(15, "qui illo error", "Baby", 623.58),
new Product(16, "aut ex ducimus", "Books", 551.39),
new Product(17, "accusamus repellendus minus", "Books", 240.58),
new Product(18, "aut accusamus quia", "Baby", 881.38),
new Product(19, "doloremque incidunt sed", "Games", 988.49),
new Product(20, "libero omnis velit", "Baby", 177.61),
new Product(21, "consectetur cupiditate sunt", "Toys", 95.46),
new Product(22, "itaque ea qui", "Baby", 677.78),
new Product(23, "non et nulla", "Grocery", 70.49),
new Product(24, "veniam consequatur et", "Books", 893.44),
new Product(25, "magnam adipisci voluptate", "Grocery", 366.13),
new Product(26, "reiciendis consequuntur placeat", "Toys", 359.27),
new Product(27, "dolores ipsum sit", "Toys", 786.99),
new Product(28, "ut hic tempore", "Toys", 316.09),
new Product(29, "quas quis deserunt", "Toys", 772.78),
new Product(30, "excepturi nesciunt accusantium", "Toys", 911.46)
);
return products;
}
```

Java 8 Excersises

- **Exercise 1** — Obtain a list of products belongs to category “Books”
- **Exercise 2** — Obtain a list of products belongs to category “Books” with price > 100
- **Exercise 3** — Obtain a list of product with category = “Toys” and then apply 10% discount
- **Exercise 4** — Get the cheapest products of “Books” category