1. **Create a class Book with members as bid,bname,price and author.Add following methods:**
   a. Constructor (Support both parameterized and parameterless)
   b. Destructor
   c. ShowBook
   d. Add static variable count and also maintain count of objects created.

```
class Book:
    count = 0  # Static variable to count objects

    def __init__(self, bid, bname, price, author):
        self.bid = bid
        self.bname = bname
        self.price = price
        self.author = author
        Book.count += 1

    def __del__(self):
        Book.count -= 1

    def Showbook(self):
        print("ID:", self.bid)
        print("Name:", self.bname)
        print("Price:", self.price)
        print("Author:", self.author)

b1 = Book(1, "Harry Potter", 500, "J.K. Rowling")
b2 = Book(2, "The Alchemist", 300, "Paulo Coelho")

b1.Showbook()
print("########################################")
b2.Showbook()
print("Total books:", Book.count)
```

2. **Create a class Product with members as pid,pname,price and quantity .Add following methods:**
   e. Constructor (Support both parameterized and parameterless)
   f. Destructor
   g. ShowBook
   h. Add static member discount.
   i. Provide methods for applying discount on price of product.

```
class Product:
    discount = 0.1  # static member (10% discount)

    def __init__(self, pid, pname, price, quantity):
        self.pid = pid
        self.pname = pname
        self.price = price
        self.quantity = quantity

    def __del__(self):
        pass

    def ShowBook(self):
        print("ID:", self.pid)
        print("Name:", self.pname)
        print("Price:", self.price)
        print("Quantity:", self.quantity)

    @staticmethod
    def set_discount(amount):
        Product.discount = amount

    def apply_discount(self):
        return self.price * (1 - Product.discount / 100)

    p1 = Product(101, "Pen", 50, 10)
    p1.ShowBook()
    print("Price after discount:", p1.apply_discount())
```

3. **Create a class Shirt with members as sid,sname,type(formal etc), price and size(small,large etc) .Add following methods:**
   j. Constructor (Support both parameterized and parameterless)
   k. Destructor
   l. ShowBook
   m. For each size of shirt price should change by 10%.

```
class Shirt:

    size_price = {"small": 1.0, "medium": 1.1,  "large": 1.2,  "xlarge": 1.3}

    def __init__(self, bid, bname, type, price, size):

        self.sid = bid

        self.sname = bname

        self.type = type

        self.price = price

        self.size = size


    def __del__(self):

        pass


    def ShowBook(self):

        print("ID:", self.sid)

        print("Name:", self.sname)

        print("Type:", self.type)

        print("Size:", self.size)


        new_price = self.price * Shirt.size_price.get(self.size.lower(), 1)

        print("Price:", new_price)
```

```
# Create shirts with different sizes
shirt1 = Shirt(1, "Shirt", "Formal", 1000, "small")
shirt2 = Shirt(2, "Shirt", "Formal", 1000, "medium")
shirt3 = Shirt(3, "Shirt", "Formal", 1000, "large")
shirt4 = Shirt(4, "Shirt", "Formal", 1000, "xlarge")

shirt1.ShowBook()
print("############")
shirt2.ShowBook()
print("############")
shirt3.ShowBook()
print("############")
shirt4.ShowBook()
```