**Singleton Design Pattern**:

We have implemented Singleton Design Pattern for Tax calculation on products.

**Problem**: To Calculate statewide tax, different value for each and every state.

**Scenario**: When Customer makes an order, the Orderdetails contains all the information regarding to the order. The Orderdetails contains all the products that are included in the order. Here each Order has a unique orderid. Every Order total should be included with certain tax amount, the tax amount varies based on customer location state.

**Solution Using Singleton design Pattern:**

We have created a new class, Tax. Here we are using a private static constructor to instantiate the object and having a getInstance() function to return that instance.

When an Order is placed, A Tax object is created, and tax is calculated for that particular order.

Tax object has only one instance and works for that customer for any number of orders made to that particular location instead of instantiating each and every time.

**Class Diagram**:

Diagram

Description automatically generated

**Collaboration Diagram**:

Diagram

Description automatically generated