

## SE 1108 PROGRAMMING AND PROBLEM SOLVING II

In this assignment, you will enhance a simple online store model that sells products divided into two categories: electronics and books. You will employ inheritance to model the relationship between the categories and their products, and abstract methods to implement tax calculations differently for each category.

Define a class called Product that represents a product for sale. Each product should have the following attributes:

- ID (integer)
- Name (string)
- Price (double)
- Description (string)
- Constructor that takes the attributes as parameters
- Getters for attributes
- An abstract method called `getTax` that calculates the tax for the product.

Define a subclass of Product called ElectronicProduct. Each ElectronicProduct should have the following attributes:

- Manufacturer (string)
- Model (string)
- Warranty (integer, representing number of years)
- Constructor that takes the attributes as parameters
- Getters for attributes
- A `getTax` method that returns 20% of the price as tax.
- A `toString` method that uses all attributes.

Define a subclass of Product called BookProduct. Each BookProduct should have the following attributes:

- Author (string)
- Genre (string)
- Number of pages (integer)
- Constructor that takes the attributes as parameters
- Getters for attributes
- A `getTax` method that returns 8% of the price as tax.
- A `toString` method that uses all attributes.

Define a class called Store that represents the online store. The store should have the following attributes:

- A list of Product objects to represent the store's inventory.
- A method to add a new product to the inventory.
- A method to remove a product from the inventory.
- A method to search for products by name and return a list of matching products.
- A method to calculate the total value of the store's inventory, including the price and tax of each product.
- A toString method.

Create a Main class that demonstrates the functionality of the Store class. In Main, create several instances of ElectronicProduct and BookProduct, add them to a Store object, and then use the Store object's methods to manipulate and display the inventory.