University of Engineering and Technology Peshawar

|  |
| --- |
|  | C:\Users\DECENT~1\AppData\Local\Temp\ksohtml4032\wps1.jpg |

**Task \_05**

Course Title: BSCS Java OOPS (lab-5)(CS-101)

**Submitted To :**

Syed Adeel Ali Shah Associate Professor CS&IT Peshawar

**Submitted by:**

Waqas khan

Registration No:24pwbcs1141

Date: 11/10 /2024

Department of CS & IT, UET Peshawar

**Lab 5 Tutorial: Advanced Inheritance Concepts**

**Assignment Description: Simple Inventory Management Application**

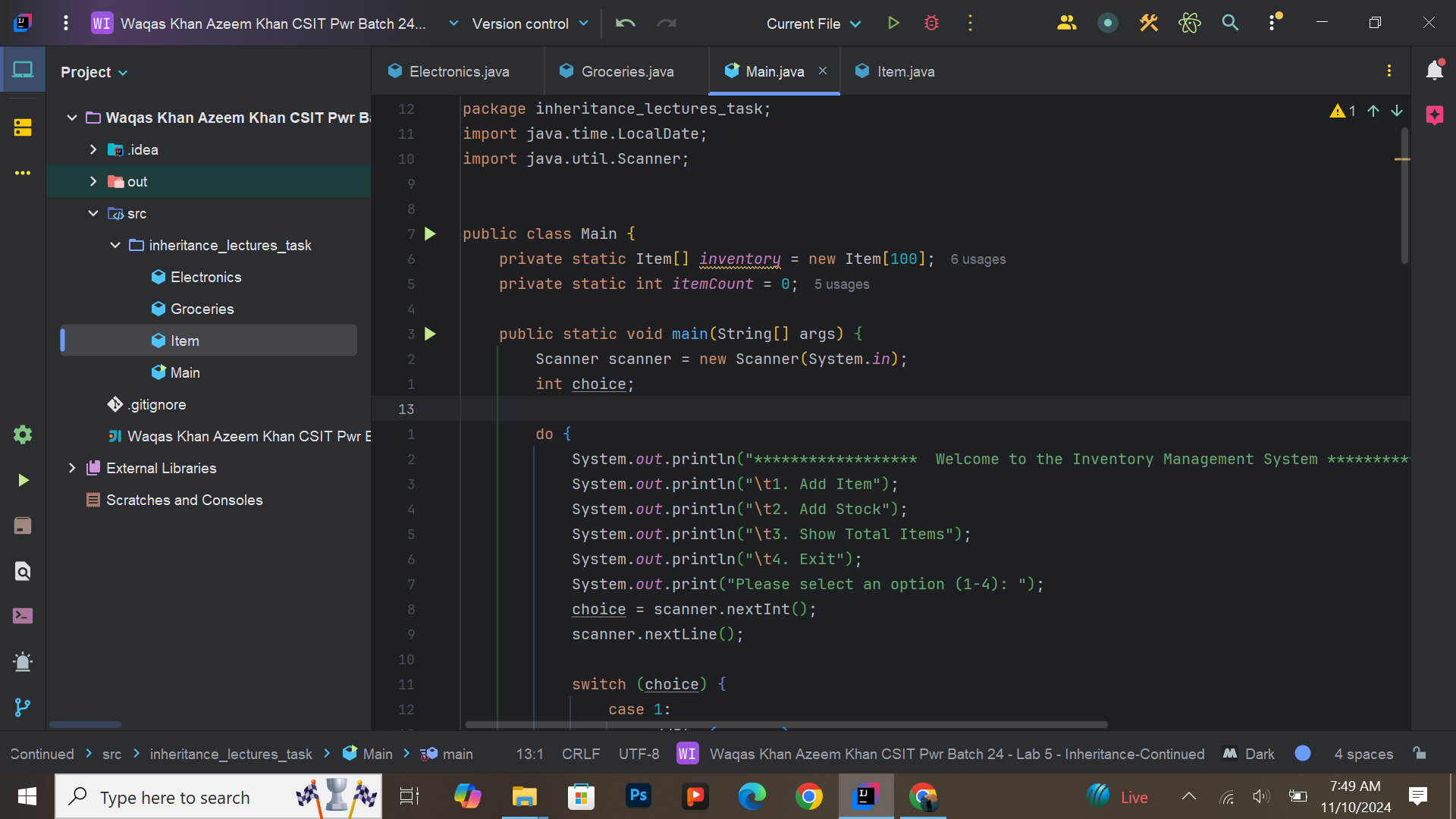
**Objective:** Create a simple inventory management application that demonstrates the concepts of inheritance, method overriding, static functions and variables, and user input using the Scanner class. The application will consist of a base class Item and derived classes for specific item types (e.g., Electronics and Groceries).

**Task Breakdown:**

1. **Classes:**
   * **Base Class:**
     + Item
       - Attributes: itemID, itemName, and quantity.
       - Methods:
         * A constructor to initialize the item details.
         * A static variable to keep track of the total number of items in the inventory.
         * A static method to display the total count of items.
         * A method to add stock to the item.
         * A method to display item details.
   * **Derived/Child Classes:**
     + Electronics - Inherits from Item and includes:
       - An additional attribute for warrantyPeriod.
       - An overridden method to add stock, which could log the addition with a message about the warranty period.
     + Groceries - Inherits from Item and includes:
       - An additional attribute for expirationDate.
       - An overridden method to add stock, which checks if the item is expired before allowing stock addition.
2. **User Input:**
   * Use the Scanner class to take input from the user for creating items in the inventory.
   * Prompt the user to enter the type of item (Electronics or Groceries), item ID, item name, and quantity.
   * After item creation, allow users to add stock to an existing item by entering the item ID and the quantity to be added.
3. **Method Overriding:**
   * Ensure that the addStock method is overridden in both derived classes to implement specific behaviors related to each item type.
4. **Static Functions and Variables:**
   * Implement a static function in the Item class to retrieve the total number of items in the inventory.
   * Use a static variable to maintain the count of items added, which should be incremented in the constructor of each item.
5. **Hints:**
   * Start by designing your classes and their relationships (inheritance).
   * Think about how to structure the addStock method for each derived class to include unique functionalities.
   * Consider how user input will flow through the program—after creating an item, what actions can the user take?
   * Ensure to validate user inputs where necessary, such as checking for valid item IDs when adding stock.

**Expected Outcome:** By the end of this assignment, students should have a functional console application that allows users to manage an inventory of different item types while applying the concepts of inheritance, method overriding, and static methods and variables in Java.

1. **Welcome Message:** The application starts with a welcome message and a menu of options.
2. **Adding Items:** When the user selects the option to add an item, they choose the type of item, input the item ID, name, and quantity, and receive a confirmation message.
3. **Adding Stock:** The user can then add stock to existing items by entering the item ID and the amount to be added, followed by a success message.
4. **Show Total Items:** The user can view the total number of unique items in the inventory.
5. **Exit Option:** The application ends gracefully with a goodbye message when the user chooses to exit.



Due to lots of lines I took a screenshot of the whole document using carban.now.sh website  
carban.now.sh : <https://carbon.now.sh>









Expected Output:

