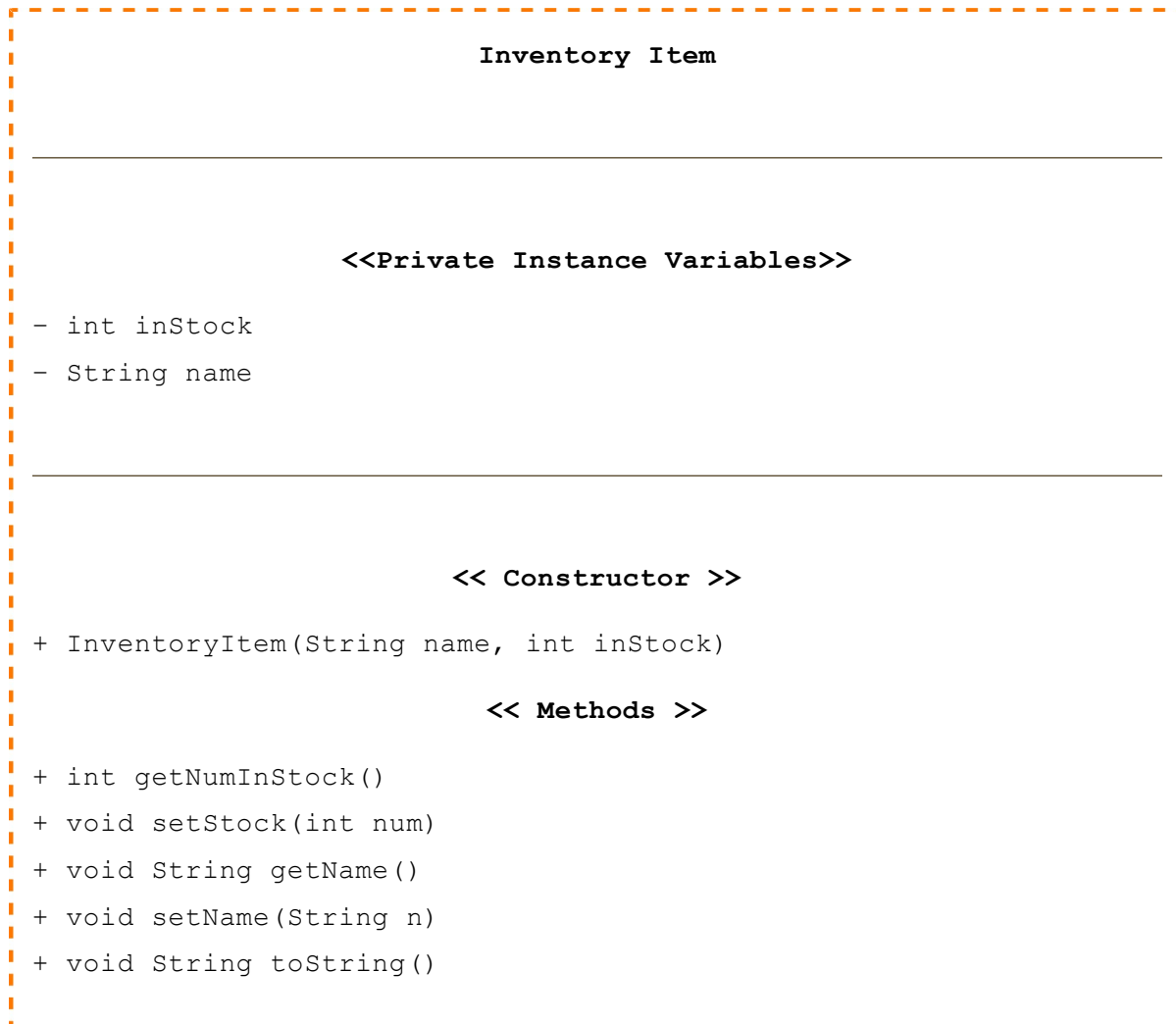# 07.04 Virtual Lecture Notes

---

Array Traversal

Traversing an array is not that difficult. You simply decide how you want to go through it (forward or backward) and then write a loop to do the job.

For this example, we'll work with objects from the InventoryItem class. The class diagram outlines the class. Compare the diagram to the class implementation.

```
                        Inventory Item
  _____

                   <<Private Instance Variables>>
  - int inStock
  - String name
  _____


                       << Constructor >>
  + InventoryItem(String name, int inStock)

                        << Methods >>
  + int getNumInStock()
  + void setStock(int num)
  + void String getName()
  + void setName(String n)
  + void String toString()
```

Now, an array of `InventoryItems` can be created. We will have just five items in our array, to keep things simple. (Notice that InventoryItem is an array of objects.)

```java
InventoryItem[] inventory = new InventoryItem[5];

// create inventory
inventory[0] = new InventoryItem("Towel", 200);
inventory[1] = new InventoryItem("Cleaning Cart", 30);
inventory[2] = new InventoryItem("Toiletry Sets", 100);
inventory[3] = new InventoryItem("Coffee Set", 300);
inventory[4] = new InventoryItem("Pillows", 50);
```

Take a look at the `printInventory` method. It is tasked with traversing the array and printing each element. The for loop keeps track of the current index position. The print statement utilizes the `toString` method of the `InventoryItems` class.

```java
public static void printInventory(InventoryItem[] itemList)
{
     for(int i = 0; i < itemList.length; i++)
          System.out.println(itemList [i]);
}
```

This method uses the traversal algorithm to go through the inventory array one item at a time, by using a traditional for loop. That is all there is to performing a traversal; use a loop and go through the array performing any action (in this case, each item).

Be sure to run the `TestInventory1` class and observe how it works. Also, try to make changes and see the result. The `findLargest` method also demonstrates traversing the array.

ArrayList Traversal

Now, how about an `ArrayList`? Take a look at the TestInventory2.java. Notice the code for performing a traversal through an ArrayList is very similar to that of the array. Run the program and observe the output. What changes would you like to implement? Give it a try.

🖨 **Print**