

16.01 Virtual Lecture Notes

Traversing through 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 example, consider an array of inventory items as defined by the following class, **InventoryItem**.

```
public class InventoryItem
{
    // instance variables
    private int inStock;
    private String name;

    /**
     * Constructor for objects of class InventoryItem
     */
    public InventoryItem(String n, int s)
    {
        // initialise instance variables
        name = n;
        inStock = s;
    }
    public int NumInStock()
    {
        return inStock;
    }
    public void setStock(int num)
    {
        inStock = num;
    }
    public String getName()
    {
        return name;
    }
    public void setName(String n)
    {
        name = n;
    }
    public String toString()
    {
        return name + ": " + inStock + " in Stock";
    }
}
```

Now, let us assume we have an array of inventory items. We will have just five items in our array, to keep things simple. (Notice that `InventoryItem` is an array of objects.)

```
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 this `printInventory()` method. Don't forget that the print statement is using the `toString()` method of the `InventoryItem` class to print the elements of the array as they are traversed.

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

- Do a desk check of the `InventoryItem.java` and `TestInventory.java` files so you know the purpose of each line of code.
- Run the files and make sure you understand them before you continue

Here is another traversal through our inventory list:

```
public static void largest(InventoryItem [] list)
{
    double max;
    int index;

    if (list.length != 0)
    {
        max = list[0].NumInStock();
        index = 0;

        for (int i = 1; i < list.length; i++)
        {
```

```
        if (max < list[i].NumInStock())
        {
            max = list [i].NumInStock();
            index = i;
        }

        System.out.println(list[index]);
        return;
    }

    System.out.println("There are no items in stock.");
}
```

Notice that this traversal determines the inventory item with the most in stock. It is a traversal, due to the for loop used to go through the inventory list one item at a time.

- Carefully study this program by doing a thorough desk check
- Run the file and make sure you understand it before you continue.

Now, how about an ArrayList? Take a look at the demo programs **TestInventory2.java** and **LargestInStock2.java**. Notice that they are the same, except for using ArrayList operations.

- Again, carefully study the code by doing a line-by-line desk check.
- Run the files and make sure you understand them before you continue.