Array review

- Can an array have data of different types?
- How do you declare a variable `isOn` to be an array that holds `true/false` values?
- Has any space in memory been made available for the array?
- When you create an array you must specify
  1.
  2.
- Can you change the values stored in the array?
- Can you change the length of an array?
- How do you get the length of an array?

Arrays of Objects

- Arrays can store references to objects in addition to primitive values.
- E.g.,
  ```java
  GiftCard[] cards = new GiftCard[4];
  ```
- Creating an array does not create the objects.
- Instead each element of the array is initialized to `null`.

```java
GiftCard[] cards = new GiftCard[4];
cards[0] = new GiftCard("Target", 25.0);
cards[1] = new GiftCard("Macy's", 50.0);
```

Creating the Objects in the Array

- To fill the array, you need to create the objects in addition to creating the array.

```java
cards[0] = new GiftCard("Target", 25.0);
cards[1] = new GiftCard("Macy's", 50.0);
```
Using an Array of Objects

• As before, you can use an an array element (array name with an index) in the same way you use a variable that is a reference to an object.

• E.g., Add $20 to the gift card at index 1.

```java
cards[1].addMoney(20.0);
```

That is, `cards[1]` is a reference to a `GiftCard` object on which to invoke the `addMoney()` method.

Arrays as a Field of a Class

• When a class has a field that is an array, do not create the array when you declare the field.

• The constructor should create the array.

```java
public class Wallet {
    GiftCard[] cards; // declare the field

    public Wallet(int numCards, String store,
                  double value) {
        cards = new GiftCard[numCards];
        for (i = 0; i < numCards; i++) {
            cards[i] = new GiftCard(store, value);
        }
    }
}
```

NullPointerException

• If program attempts to use `null` when an object is required, Java throws a `NullPointerException`:

  • Invoke a method of a `null` object:
    ```java
cards[3].buyGoods(30.0); (Error)
```

  • Access or modify a field of a `null` object:
    ```java
    b = cards[2].balance; (Error)
    ```

  • Access length or element of a `null` as if it were an array:
    ```java
    int[] data;
    if (data.length > 0) (Error)
    ```