Array Review

• Arrays are a fixed size, collection of values of the same type.
• Arrays are indexed starting at 0.
• You can find the length of an array by calling array.length.
• Creating an array:
  Object arrayName = new Object[size];
  Object arrayName = {obj1, obj2, ...};
Array Review (2)

• Accessing arrays:
  arrayName[indexPosition]
• Copy arrays using
  System.arraycopy(arrayFrom, positionFrom, arrayTo, positionTo, count);
• Arrays may be used as parameters, return values and instance variables
• Command line arguments are stored in a String array.

Array Review (3)

• Use partially filled arrays to emulate a variable sized array
  – Then you can add (as long as there is room) and remove elements
• 2D arrays are used to model tables and matrices
• Creating a 2D array:
  Object[][] arrayName2D = new Object[rows][columns];
Array Review (4)

- Parallel arrays are used to store data that corresponds at the $i^{th}$ element
  - Difficult to maintain
- Use a container class to hold each of the objects needed in a parallel array
  - Instead of having a title and artist array have a Song array

Vectors

- Vectors are an object container that can grow
  - You don’t need to create a new container of a larger size
- Import the java.util package
Creating a Vector

• There are four Vector constructors
  – Vector()
    • Sets the size to 10 and the capacity increment to zero
  – Vector(Collection c)
    • Creates a vector with the elements of the Collection
  – Vector(int initialCapacity)
    • Creates a vector with the specified capacity
  – Vector(int initialCapacity, int capacityIncrement)
    • Creates a vector with the specified capacity and capacity increment

Adding an Object to a Vector

• Use the add method
  Vector songs = new Vector();
  Song s1 = new Song(“song1”, “artist1”);
  Song s2 = new Song(“song2”, “artist2”);
  songs.add(s1); //Adds to position 0
  songs.add(s2); //Adds to position 1
Replacing an Object in a Vector

- Use the set method to specify what position to put an object in

```java
Vector songs = new Vector();
Song s1 = new Song("song1", "artist1");
Song s2 = new Song("song2", "artist2");
songs.add(s1); //Adds to position 0
songs.set(0, s2); //Replaces s1 with s2
```

Removing an Object from a Vector

- Use remove(int index) to remove an object from a specific index
- Use remove(Object o) to remove the first instance of a specific object
Removing an Object from a Vector

Vector songs = new Vector();
Song s1 = new Song(“song1”, “artist1”);
Song s2 = new Song(“song2”, “artist2”);
songs.add(s1); //Adds to position 0
songs.add(s2); //Adds to position 1
songs.remove(1); //Removes s2
songs.remove(s1); //Removes s1

Size of the Vector

• Use the size() method to obtain the size of the vector (the number of elements)
Vector songs = new Vector();
Song s1 = new Song(“song1”, “artist1”);
Song s2 = new Song(“song2”, “artist2”);
songs.add(s1); //Adds to position 0
songs.add(s2); //Adds to position 1
int size = songs.size(); //Size is 2
Getting an Element from the Vector

• Use the get(int index) to get an element from a vector

```
Vector songs = new Vector();
Song s1 = new Song("song1", "artist1");
Song s2 = new Song("song2", "artist2");
songs.add(s1); //Adds to position 0
songs.add(s2); //Adds to position 1
for(int x = 0; x < songs.size(); x++) {
    Song current = (Song)songs.get(x);
    System.out.println(current);
}
```

References

• Jason Schwarz’s Lecture 24 & 25 slides: http://courses.ncsu.edu/csc116/