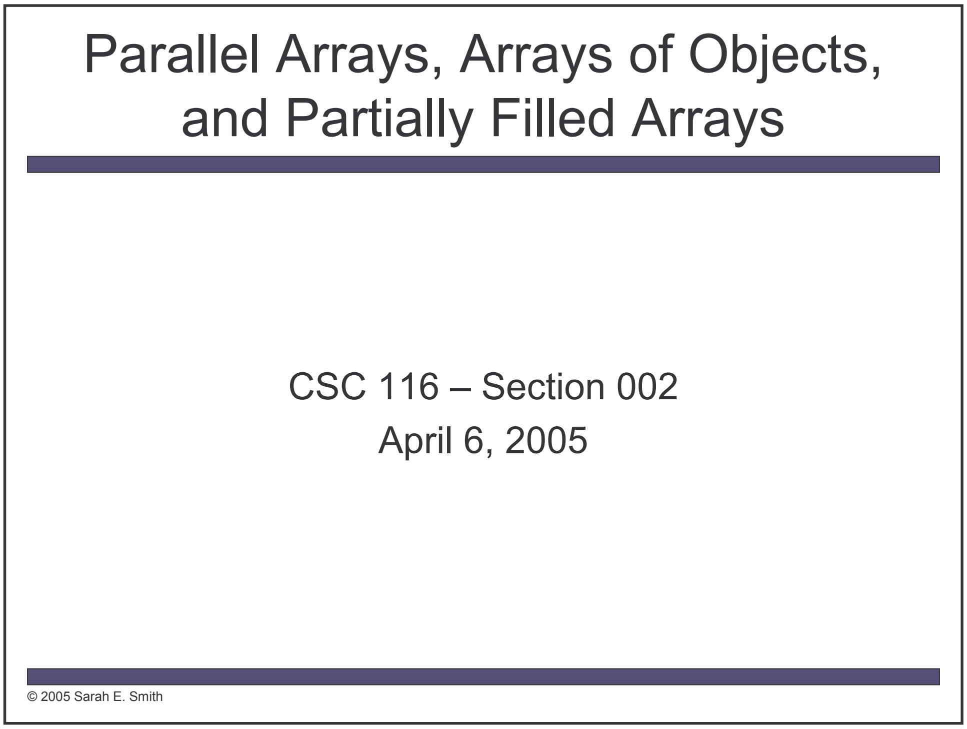
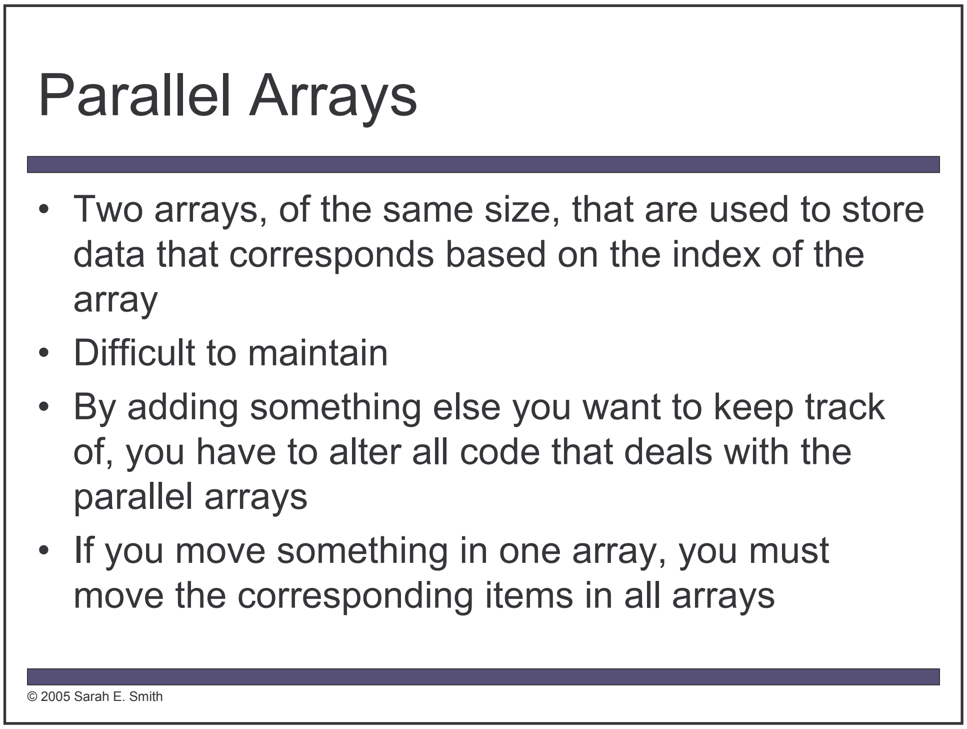


# Parallel Arrays, Arrays of Objects, and Partially Filled Arrays

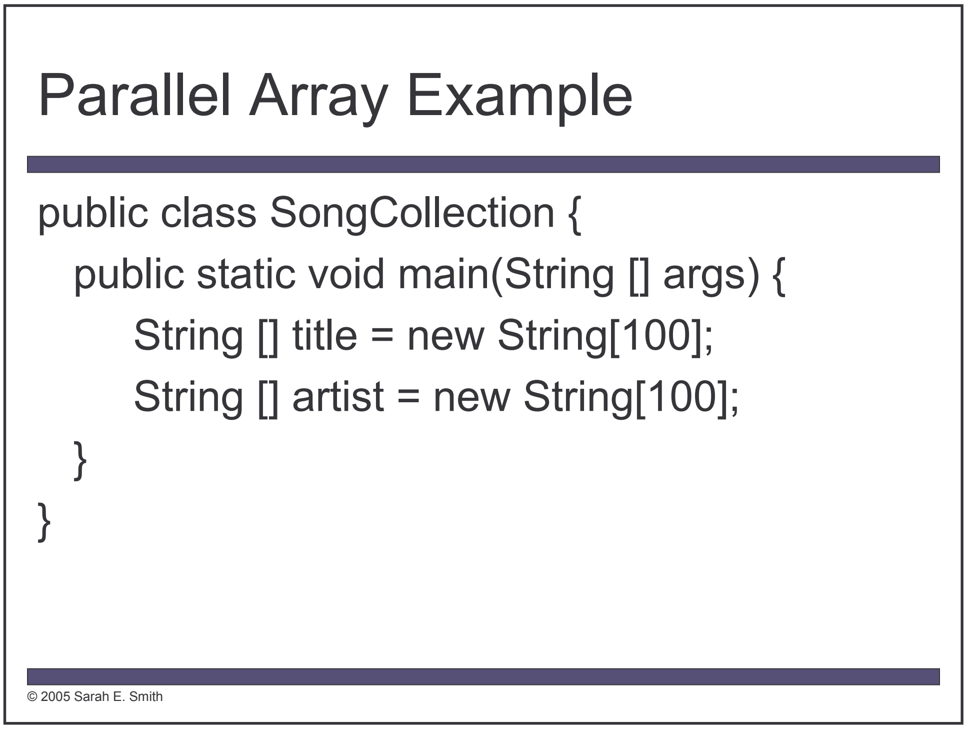
CSC 116 – Section 002

April 6, 2005



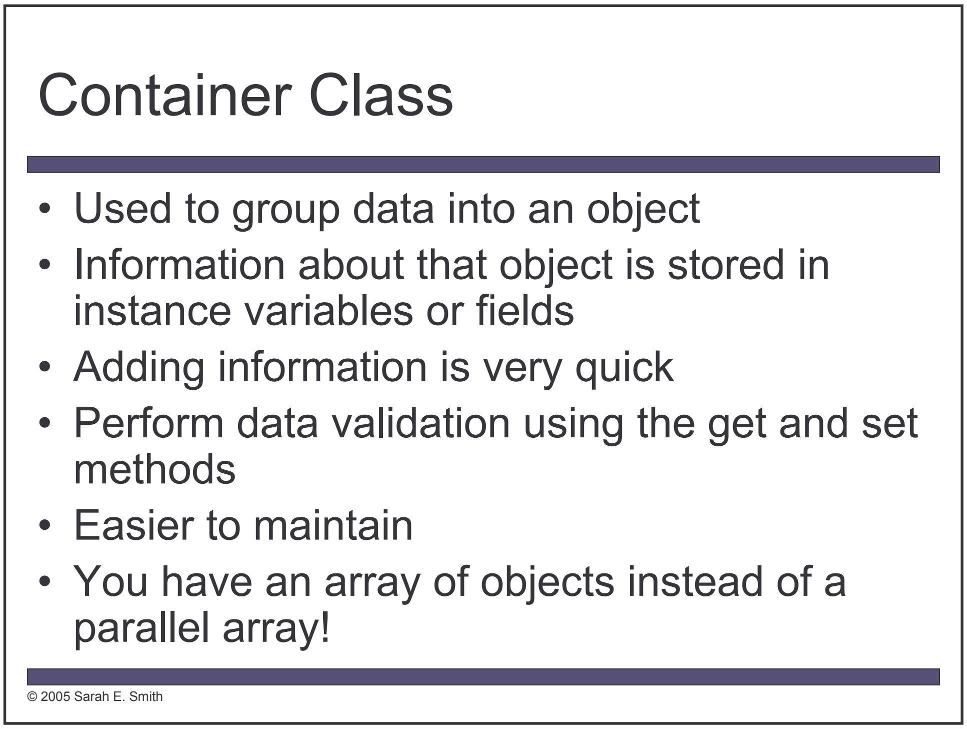
## Parallel Arrays

- Two arrays, of the same size, that are used to store data that corresponds based on the index of the array
- Difficult to maintain
- By adding something else you want to keep track of, you have to alter all code that deals with the parallel arrays
- If you move something in one array, you must move the corresponding items in all arrays



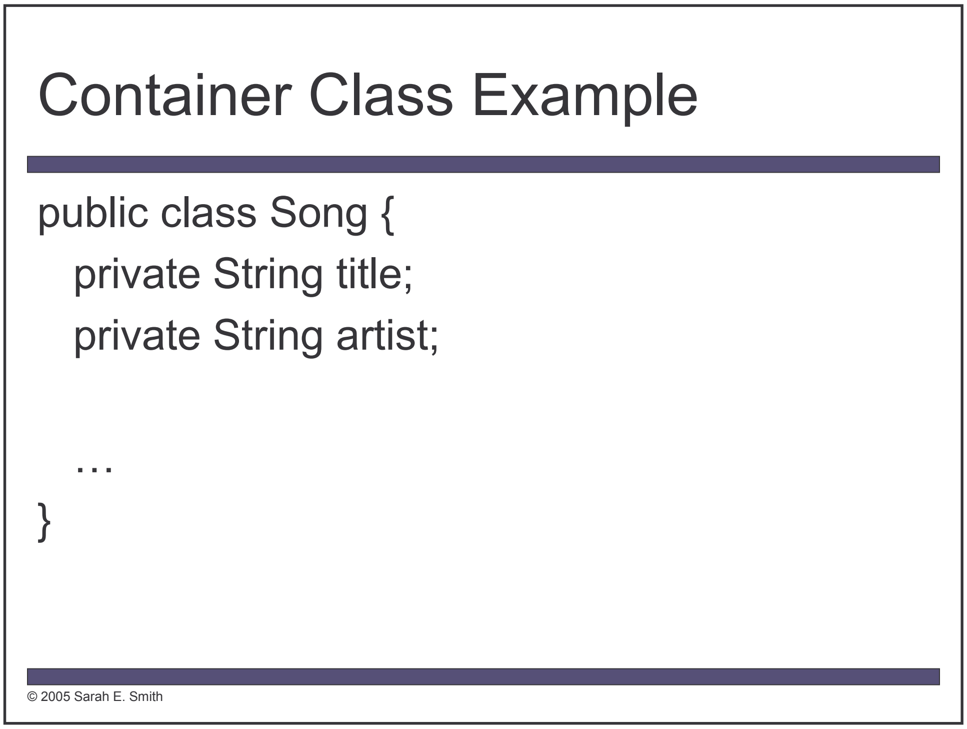
## Parallel Array Example

```
public class SongCollection {
   public static void main(String [] args) {
        String [] title = new String[100];
        String [] artist = new String[100];
   }
}
```



## Container Class

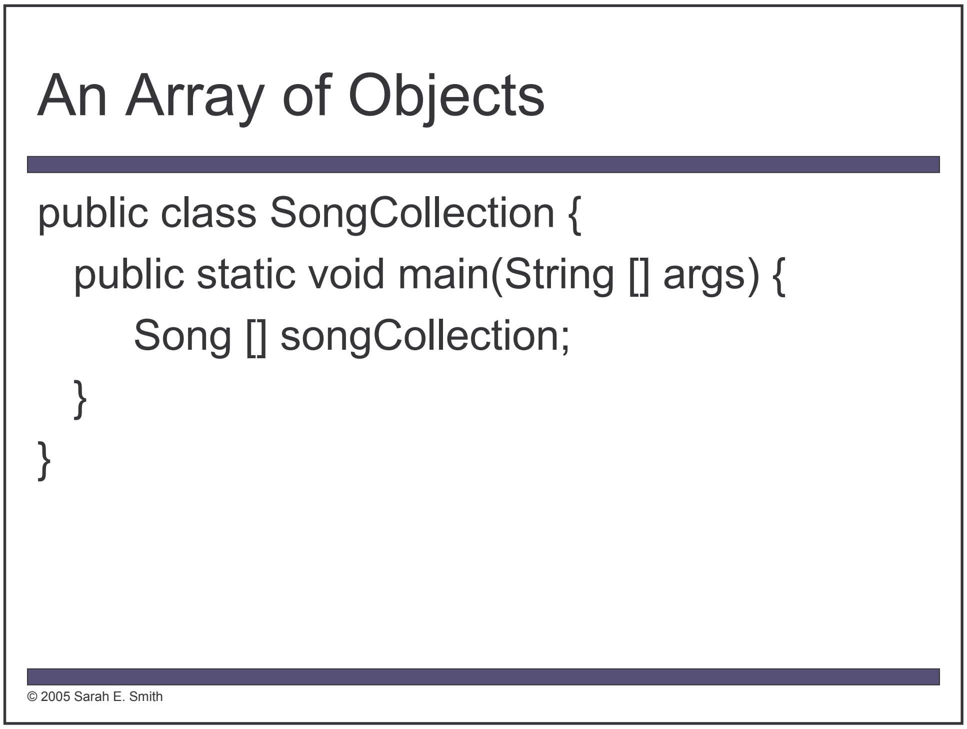
- Used to group data into an object
- Information about that object is stored in instance variables or fields
- Adding information is very quick
- Perform data validation using the get and set methods
- Easier to maintain
- You have an array of objects instead of a parallel array!



## Container Class Example

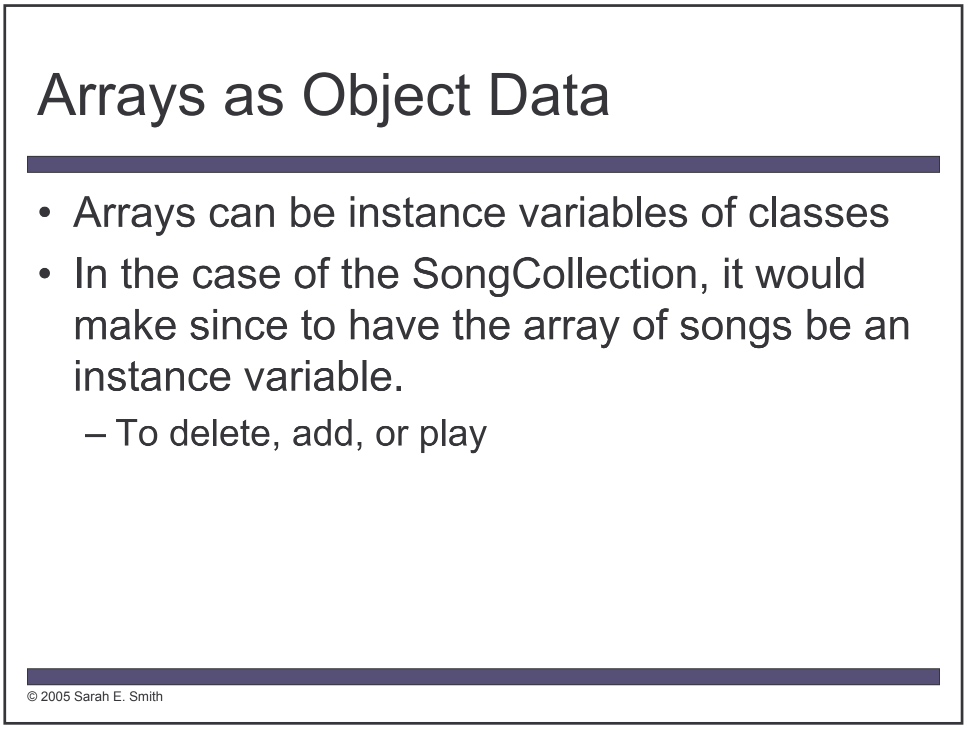
```
public class Song {
   private String title;
   private String artist;

   …
}
```



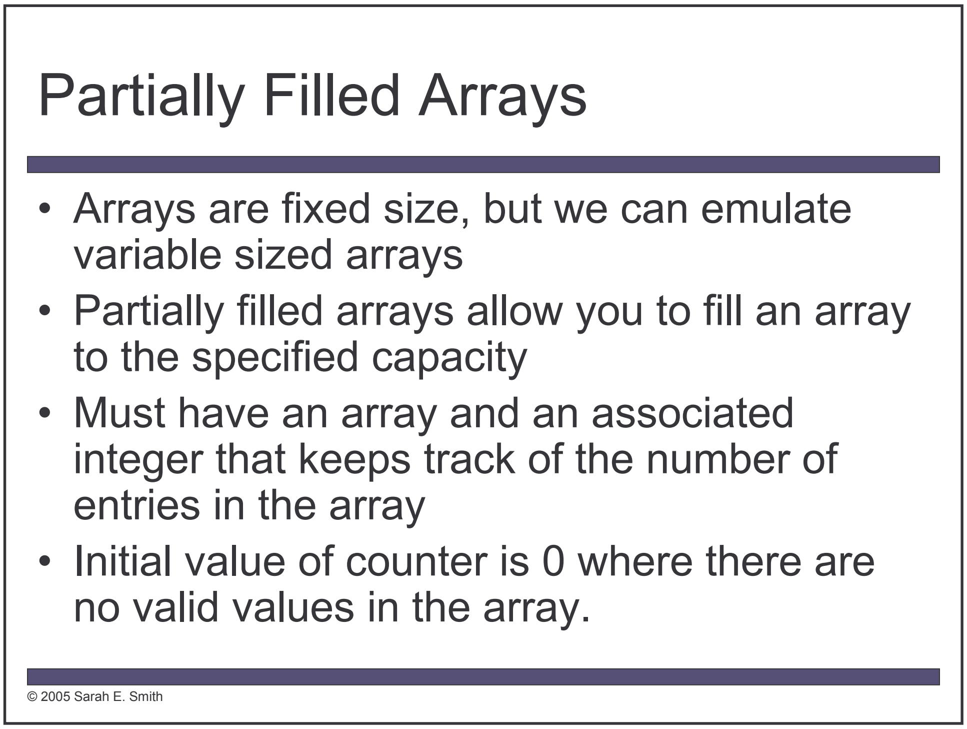
## An Array of Objects

```
public class SongCollection {
   public static void main(String [] args) {
        Song [] songCollection;
   }
}
```



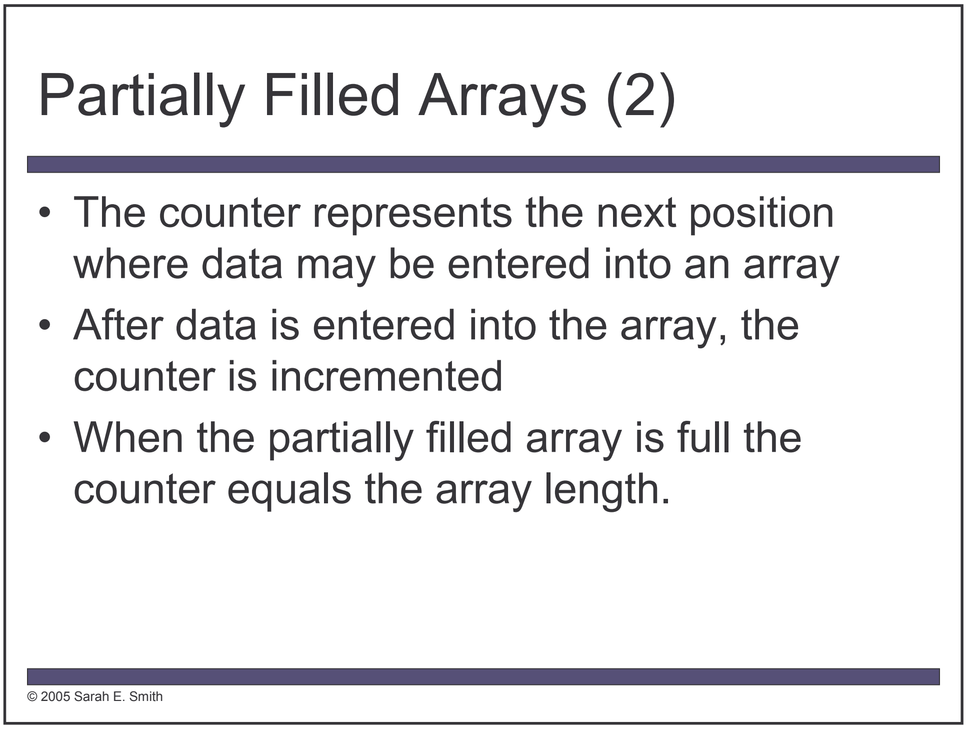
## Arrays as Object Data

- Arrays can be instance variables of classes
- In the case of the SongCollection, it would make since to have the array of songs be an instance variable.
  - To delete, add, or play



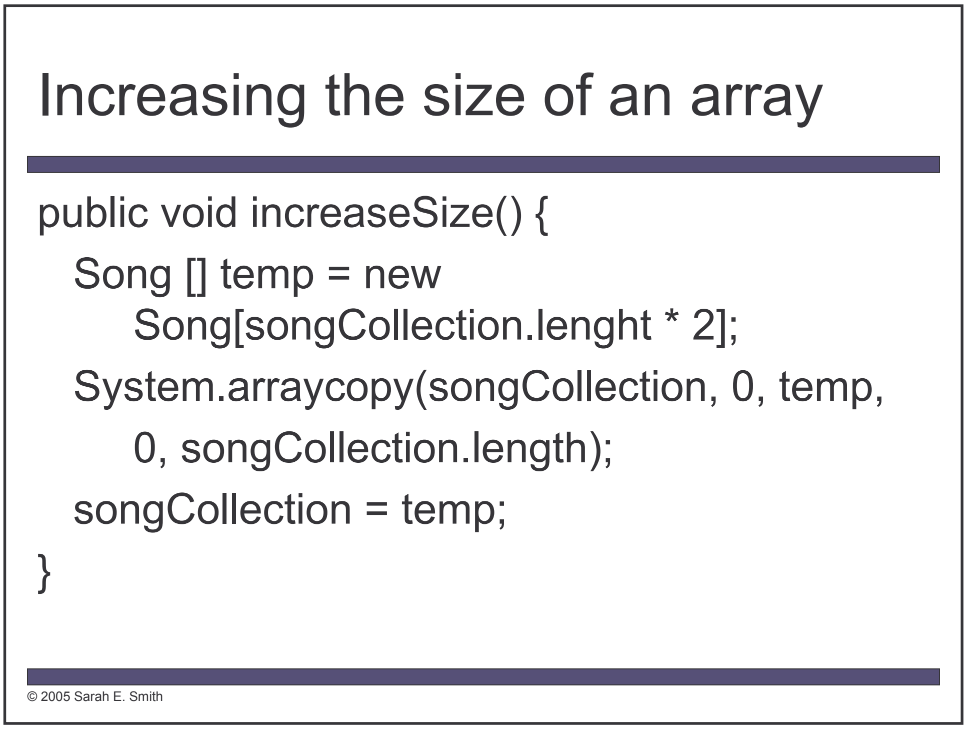
## Partially Filled Arrays

- Arrays are fixed size, but we can emulate variable sized arrays
- Partially filled arrays allow you to fill an array to the specified capacity
- Must have an array and an associated integer that keeps track of the number of entries in the array
- Initial value of counter is 0 where there are no valid values in the array.



## Partially Filled Arrays (2)

- The counter represents the next position where data may be entered into an array
- After data is entered into the array, the counter is incremented
- When the partially filled array is full the counter equals the array length.



## Increasing the size of an array

```
public void increaseSize() {
   Song [] temp = new
        Song[songCollection.lenght * 2];
   System.arraycopy(songCollection, 0, temp,
        0, songCollection.length);
   songCollection = temp;
}
```

# References

- Jason Schwarz's Lecture 23 slides: http://courses.ncsu.edu/csc116/
- Carol Miller's Lecture 11 slides