Today

Lists: ordered collection of objects

- Operators: insertion, deletion, search
- Representation: linked lists versus arrays
- Performance implications

Project 3: ArrayLists and binary I/O

CS 2334: Project 3

Objectives:

- Use ArrayLists: constructing, sorting, searching, iterating
- Storing objects in and retrieving objects from binary files
  - Serializable objects
  - ObjectOutputStream / ObjectInputStream

Milestone 1: ArrayLists

ArrayList implementation of FinchActionList:

```
public class FinchActionList extends
  ArrayList<FinchAction> implements Serializable{...}
```

- Provides: add, get, iteration
- Collections provides: sorting, reversal
- (Serialization key in latter milestones)

Milestone 2: Execution/Display in Natural and Reverse Order

Update FinchActionList methods:

```
public void display(String name, boolean reverse)
public void execute(Finch myFinch, String name, boolean reverse)
```

- Reverse = false : natural order
- Reverse = true : reverse order

- User command: specify natural or reverse order

Milestone 3: Write User Command

Add to your driver class:

```
public static void WriteList(String fname,
  FinchActionList list, String action_name);
```

- ObjectOutputStream will allow you to write an entire FinchActionList to a file in one call to the writeObject() method
- Possible because the FinchActionList is Serializable
- … more details on Friday

Milestone 4: Add Read and Merge User Commands

Add to your driver class:

public static FinchActionList ReadList(String fname);

- Read a FinchActionList from a file
- Read command: replace the current list
- Merge command: add the new list to the current list
  - Remember to re-sort after the merge

Milestone 5: Add the Clean User Command

- Remove the duplicate entries from the current list
- Assume that the list is already sorted:
  – Duplicate entries are next to each other in the list
  – We will now outline the algorithm on the board…

One More Note…

- Make sure that all class constructors check for valid parameters
- For the purposes of this project, if a parameter is not within the valid range, then set it to a reasonable default.

Deadlines

- October 15th @ 5:00pm: design
- October 22nd @ 5:00pm : final version, including demonstration