

Mat 2170  
Week 14

**ArrayList**  
Class

Week14

Lab 14

ArrayList  
Class

Generic Types

Wrapper  
Classes

Search

# Mat 2170 Week 14

## ArrayList Class

Spring 2014

# Student Responsibilities

Mat 2170  
Week 14

ArrayList  
Class

Week14

Lab 14

ArrayList  
Class

Generic Types

Wrapper  
Classes

Search

- **EXAM** – Thursday, 4/24, 7:00 pm  
one sheet of 8.5" by 11" paper for notes is allowed
- Reading: Textbook, Chapter 11
- Lab: ArrayList and more on writing classes from scratch
- Attendance

# Lab 14: One Handed Solitaire – An OverView

Mat 2170  
Week 14

ArrayList  
Class

Week14

Lab 14

ArrayList  
Class

Generic Types

Wrapper  
Classes

Search

- A standard **Deck** of 52 cards, shuffled (with user's seed)
- Play continues until **Deck** is empty
- When **Hand** is empty, deal from "top," otherwise deal from "bottom"
- After a **Card** is dealt, "collapse" **Hand** (if possible), comparing the "top" **Card**, and the one 3 **Cards** back from it
  - If the **Ranks** match, discard the 4 top **Cards** in the **Hand**
  - If the **Suits** match, discard the two **Cards** after the top **Card**
- Continue collapsing until no more matches
- Score is number of **Cards** left in **Hand** at the end of the game
- Program repeats until user enters -1 for the shuffle seed

# Class Card

Mat 2170  
Week 14

ArrayList  
Class

Week14

Lab 14

ArrayList  
Class

Generic Types

Wrapper  
Classes

Search

- Data members:
  - suit
  - rank
  - (faceup isn't needed for this game)
- Member methods:
  - constructor()
  - suitsMatch(), ranksMatch()
  - toString(), quickString()

# Class Deck

Mat 2170  
Week 14

ArrayList  
Class

Week14

Lab 14

ArrayList  
Class

Generic Types

Wrapper  
Classes

Search

- Data member:
  - an ArrayList of Card
  
- Member methods:
  - Two constructors
  - shuffle(seed)
  - isEmpty(), size()
  - dealTop(), dealBottom()
  - getCard(), add(), remove()
  - toString()

# The Deck Shuffle

Mat 2170  
Week 14

ArrayList  
Class

Week14

Lab 14

ArrayList  
Class

Generic Types

Wrapper  
Classes

Search

```
public void shuffle(int seed)
{
    Collections.shuffle(deck, new Random(seed));
}
```

Where `deck` is the name of the `ArrayList` in `Deck` class.

Use: `import java.util.*`

# The Collapse

Mat 2170  
Week 14

ArrayList  
Class

Week14

Lab 14

ArrayList  
Class

Generic Types

Wrapper  
Classes

Search

If there are at least 4 cards in the Hand:

    Create and initialize currentCard (on top)  
    and matchCard (3 back)

While there are at least 4 cards in the Hand,  
and either ranks or suits match:

    If ranks match,  
        delete the top 4 cards  
    Else if suits match,  
        delete the 2 below the top card

    If Hand has at least 4 cards  
        re-initialize currentCard and matchCard

# Lab 14

Mat 2170  
Week 14

**ArrayList**  
Class

Week14

**Lab 14**

ArrayList  
Class

Generic Types

Wrapper  
Classes

Search

Questions?



# The ArrayList Class

Mat 2170  
Week 14

ArrayList  
Class

Week14

Lab 14

ArrayList  
Class

Generic Types

Wrapper  
Classes

Search

- The **java.util** package includes a class called **ArrayList** that extends the usefulness of arrays by providing additional operations.
- Since **ArrayList** is a class, all operations on **ArrayList** objects are indicated using method calls.
- In the summary of **ArrayList** methods which follows, the notation **<T>** indicates the **base type** of the ArrayList object.

# ArrayList Methods

Mat 2170  
Week 14

ArrayList  
Class

Week14

Lab 14

ArrayList  
Class

Generic Types

Wrapper  
Classes

Search

```
boolean add(<T> element)
```

Adds a new element to the end of the ArrayList;  
the return value is always true

```
void add(int index, <T> element)
```

Inserts a new element into the ArrayList;  
**before** the position specified by index

```
<T> remove(int index)
```

Removes the element at the specified position and  
returns that value

```
boolean remove(<T> element)
```

Removes the first instance of element, if it appears;  
returns true if a match is found

```
void clear()
```

Removes all elements from the ArrayList

```
int size()
```

Returns the number of elements in the ArrayList

```
<T> get(int index)
```

Returns the object at the specified index

```
<T> set(int index, <T> value)
```

Sets the element at the specified index to the new value and returns the old value

```
indexOf(<T> value)
```

Returns the index of the first occurrence of the specified value, or  $-1$  if it does not appear

```
boolean contains(<T> value)
```

Returns true if the ArrayList contains the specified value

```
boolean isEmpty()
```

Returns true if the ArrayList contains no elements

# Generic Types in Java

Mat 2170  
Week 14

ArrayList  
Class

Week14

Lab 14

ArrayList  
Class

Generic Types

Wrapper  
Classes

Search

- The **type parameter** `< T >` used in the previous slides is a placeholder for the **element type** used in the array.
- Class definitions that include a **type parameter** are called **generic** types.
- When we declare or create an ArrayList, it is a good idea to specify the element type in **angle brackets**. For example:

```
ArrayList<String> myNames =  
    new ArrayList<String>();
```

- This allows Java to check for the correct element type when `set()` is called, and eliminates the need for a type cast when `get()` is called.

- Java includes a wrapper class to correspond to each of the primitive types:

<code>boolean</code>	<code>↔</code>	<code>Boolean</code>	<code>float</code>	<code>↔</code>	<code>Float</code>
<code>byte</code>	<code>↔</code>	<code>Byte</code>	<code>int</code>	<code>↔</code>	<code>Integer</code>
<code>char</code>	<code>↔</code>	<code>Character</code>	<code>long</code>	<code>↔</code>	<code>Long</code>
<code>double</code>	<code>↔</code>	<code>Double</code>	<code>short</code>	<code>↔</code>	<code>Short</code>

- The value stored in the object `maxItems` is an object, and we can use it in any context that require objects.

# Using Wrapper Classes

Mat 2170  
Week 14

ArrayList  
Class

Week14

Lab 14

ArrayList  
Class

Generic Types

Wrapper  
Classes

Search

- All of the primitive wrapper classes in Java are **immutable** – their states cannot be modified after they are created.

- For each wrapper class, Java defines a method to retrieve the primitive value, e.g.:

```
int underlyingValue = maxItems.intValue();
```

- Java will automatically **box** and **unbox** the primitive values in a wrapper class.

# Generic Types and Boxing/Unboxing

Mat 2170  
Week 14

ArrayList  
Class

Week14

Lab 14

ArrayList  
Class

Generic Types

Wrapper  
Classes

Search

- Automatic conversion of values between a primitive type and the corresponding wrapper class allows an ArrayList object to store primitive values, even though the elements of any ArrayList must be a Java class.
- For example:

```
ArrayList <Integer> myList = new ArrayList<Integer>();  
myList.add(42);  
int answer = myList.get(0);
```

In the second statement, Java uses **boxing** to enclose 42 in a wrapper object of type Integer; the third statement **unboxes** the Integer to obtain the int.



# Reversing an ArrayList

Mat 2170  
Week 14

ArrayList  
Class

Week14

Lab 14

ArrayList  
Class

Generic Types

Wrapper  
Classes

Search

```
import acm.program.*;
import java.util.*;

public class ReverseArrayList extends ConsoleProgram {
    public void run()
    {
        println("This program reverses the elements " +
                "in an ArrayList.");
        println("Use " + SENTINEL + " to signal the " +
                "end of the list.");

        ArrayList<Integer> myList = readIntArrayList();
        reverseArrayList(myList);
        printIntArrayList(myList);
    }
}
```

## readIntArrayList()

```
/* Reads the data into the list */
private ArrayList<Integer> readIntArrayList()
{
    ArrayList<Integer> list = new ArrayList<Integer>();

    int value = readInt(" ? ");
    while (value != SENTINEL)
    {
        list.add(value);
        value = readInt(" ? ");
    }

    return list;
}

/* Private constant --- Define the end-of-data value */
private static final int SENTINEL = 0;
```

Mat 2170  
Week 14

ArrayList  
Class

Week14

Lab 14

ArrayList  
Class

Generic Types

Wrapper  
Classes

Search

## reverseArrayList() & swapElements()

```
/* Reverses the data in an ArrayList */
private void reverseArrayList(ArrayList<Integer> list)
{
    for (int i = 0; i < list.size() / 2; i++)
    {
        swapElements(list, i, list.size() - i - 1);
    }
}

/* Exchanges two elements in an ArrayList */
private void swapElements(ArrayList<Integer> list,
                           int p1, int p2)
{
    int temp = list.get(p1);
    list.set(p1, list.get(p2));
    list.set(p2, temp);
}
```

Mat 2170  
Week 14

ArrayList  
Class

Week14

Lab 14

ArrayList  
Class

Generic Types

Wrapper  
Classes

Search

# ArrayList Searching Methods

Mat 2170  
Week 14

ArrayList  
Class

Week14

Lab 14

ArrayList  
Class

Generic Types

Wrapper  
Classes

Search

Method	Description
<code>contains(value)</code>	returns true if the given value appears in the list  Ex: <code>list.contains("hello")</code>
<code>indexOf(value)</code>	returns the index of the first occurrence of the given value in the list (-1 if not found)  Ex: <code>list.indexOf("world")</code>
<code>lastIndexOf(value)</code>	returns the index of the last occurrence of the given value in the list (-1 if not found)  Ex: <code>list.lastIndexOf("hello")</code>

Where **list** is an `ArrayList<string>`.

# ArrayList Sorting and Binary Search Methods

The **java.util** package contains a class called **Collections** which contains several useful static methods. Of particular interest:

Method	Description
<code>sort(list)</code>	rearranges the elements into sorted (non-decreasing) order Ex: <code>Collections.sort(L)</code>
<code>binarySearch(list, value)</code>	searches a <b>sorted</b> list for a given element value and returns its index Ex: <code>Collections.binarySearch(L, "hello")</code>

When an **ArrayList** is sorted, **binary search** is **much** faster than **linear search**.

Mat 2170  
Week 14

ArrayList  
Class

Week14

Lab 14

ArrayList  
Class

Generic Types

Wrapper  
Classes

Search