

Collections

Michael Wagner

List

```
import java.util.*;

public class CollectionDemo
{
    public static void main(String[] args)
    {
        List<String> stuff = new ArrayList<String>();
        stuff.add("Los Angeles");
        stuff.add("Ventura");
        stuff.add("Oxnard");
        stuff.add("Agoura Hills");
        stuff.add("Van Nuys");
        stuff.add("Ventura");

        for (int subscript = 0; subscript < stuff.size(); subscript++)
        {
            String current = stuff.get(subscript);
            System.out.println(current);
        }
    }
}
```

For-Each

```
import java.util.*;

public class CollectionDemo
{
    public static void main(String[] args)
    {
        List<String> stuff = new ArrayList<String>();
        stuff.add("Los Angeles");
        stuff.add("Ventura");
        stuff.add("Oxnard");
        stuff.add("Agoura Hills");
        stuff.add("Van Nuys");
        stuff.add("Ventura");

        for (String current : stuff)
        {
            System.out.println(current);
        }
    }
}
```

Example

Load the movie database on the course site.
Determine the number of movies in that file.

Set

```
import java.util.*;

public class CollectionDemo
{
    public static void main(String[] args)
    {
        Set<String> stuff = new TreeSet<String>();
        stuff.add("Los Angeles");
        stuff.add("Ventura");
        stuff.add("Oxnard");
        stuff.add("Agoura Hills");
        stuff.add("Van Nuys");
        stuff.add("Ventura");

        for (String current : stuff)
        {
            System.out.println(current);
        }
    }
}
```

Example

Load the movie database on the course site.
Find the number of unique movie titles.

```
public class Movie implements Comparable<Movie>
```

Map

```
import java.util.*;

public class CollectionDemo
{
    public static void main(String[] args)
    {
        Map<String, Integer> stuff = new HashMap<String, Integer>();
        stuff.put("Los Angeles", 3884000);
        stuff.put("Ventura", 108817);
        stuff.put("Oxnard", 203007);
        stuff.put("Agoura Hills", 20681);
        stuff.put("Van Nuys", 136443);
        stuff.put("Ventura", 110000);

        for (String key : stuff.keySet())
        {
            int value = stuff.get(key);
            System.out.println("Population of " + key + " is " + value);
        }
    }
}
```

Stack

```
import java.util.*;

public class CollectionDemo
{
    public static void main(String[] args)
    {
        Stack<String> stuff = new Stack<String>();
        stuff.push("Los Angeles");
        stuff.push("Ventura");
        stuff.push("Oxnard");
        stuff.push("Agoura Hills");
        stuff.push("Van Nuys");
        stuff.push("Ventura");

        while (!stuff.isEmpty())
        {
            String current = stuff.pop();
            System.out.println(current);
        }
    }
}
```


For next slide

```
public class City implements Comparable<City>
{
    private String name;
    private int population;

    public City(String name, int population)
    {
        this.name = name;
        this.population = population;
    }

    public int compareTo(City other_city)
    {
        if (population < other_city.population)
            return -1;
        else if (population == other_city.population)
            return 0;
        else
            return 1;
    }

    public String toString()
    {
        return name + " has a population of " + population;
    }
}
```

Queue

```
import java.util.*;

public class CollectionDemo
{
    public static void main(String[] args)
    {
        Queue<City> stuff = new PriorityQueue<City>();
        stuff.add(new City("Los Angeles", 3884000));
        stuff.add(new City("Ventura", 108817));
        stuff.add(new City("Oxnard", 203007));
        stuff.add(new City("Agoura Hills", 20681));
        stuff.add(new City("Van Nuys", 136443));
        stuff.add(new City("Ventura", 110000));

        while (!stuff.isEmpty())
        {
            City current = stuff.poll();
            System.out.println(current);
        }
    }
}
```

Example

Load the movie database on the course site.
Order the movies by year.