

# Welcome to CSE 143!



# Context for CSE 143

## **CSE 142**

- Control: loops, if/else, methods, parameters, returns
- I/O: Scanners, user input, files
- Data: primitive types (int, double, etc.), *arrays, classes*

## **CSE 143**

- Control: recursion
- Data
  - Java collections
  - Classes + Object Oriented Programming
- Best of CS

# Collections

- **collection**: an object that stores data; a.k.a. "data structure"
  - the objects stored are called **elements**
  - some collections maintain an ordering; some allow duplicates
  - typical operations: *add, remove, clear, contains* (search), *size*
- examples found in the Java class libraries: (covered in this course!)
  - `ArrayList, LinkedList, HashMap, TreeSet, PriorityQueue`
- all collections are in the `java.util` package

```
import java.util.*;
```

# Client - Radio



# Implementer - Radio



# Client – ArrayList

```
ArrayList<String> list:  
  ["a", "b", "c"]
```

# Implementer - ArrayList

String[] elementData:

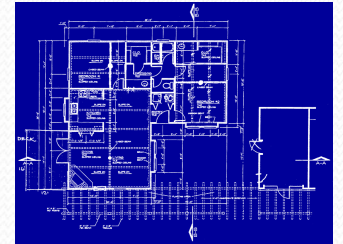
```
["a", "b", "c", null, null, null, null, null, null]
```

int size:

3

# Recall: classes and objects

- **class**: A program entity that represents:
  - A complete program or module, or
  - A template for a type of objects.
  - (`ArrayList` is a class that defines a type.)



- **object**: An entity that combines **state** and **behavior**.
  - **object-oriented programming (OOP)**: Programs that perform their behavior as interactions between objects.
  - **abstraction**: Separation between concepts and details. Objects provide abstraction in programming.