Data Structures

“A data structure is a specialized format for organizing, processing, retrieving and storing data. There are several basic and advanced types of data structures, all designed to arrange data to suit a specific purpose. Data structures make it easy for users to access and work with the data they need in appropriate ways.” ~David Loshin & Sarah Lewis

Collection

A data structure which allows a variety of like objects to be dealt with together. Examples we will use in Java:

- Array
- List
- Stack
- Queue
- Map

Array

A sequence of contiguous objects in a fixed size, can be accessed by an index.

```
String[] names = new String[] { "Jed", "Amy", "Rafael", "Lin", "Oskar" };
names[0] = “Jedi”;
System.out.println(“First element is “ + names[0]);
System.out.println(“Last element is “ + names[names.length-1]);

for (int i = 0; i < names.length; i++) {
        // act on element at index i, and can also use i
}

for (String name : names) {
        // act on each name element, but no access to index
}
```

List

A sequence of objects, can be accessed by an index.

```
List<String> names = new ArrayList<String>(Arrays.asList( "Jed", "Amy",
"Rafael", "Lin", "Oskar"));
list.set(0, "Jedi");
System.out.println(“First element is “ + names.get(0));
System.out.println(“Last element is “ + names.get(names.size()-1));

for (int i = 0; i < names.size(); i++) {
        // act on element at index i using get, and can also use i
}

for (String name : names) {
        // act on each name, but no access to index
}
```

Stack

```
Stack<String> names = new Stack<String>();
names.push("Jed");
System.out.println(“Peek at top element: “ + names.peek());
System.out.println(“Pop top element off stack + names.pop());

while (names.size() > 0) {
        System.out.println(names.pop());
}

// Looping through the stack should seldom be used, as it violates the intent
for (String name : names) {
        // act on each name, but no access to index
}
```

Queue

```
Queue<String> names = new LinkedList<String>();
names.offer("Jed");
System.out.println(“Get oldest element off queue + names.poll());

while (names.size() > 0) {
        System.out.println("Next: "+names.poll());
}

// Looping through the queue should seldom be used, as it violates the intent
for (String name : names) {
        // act on each name, but no access to index
}
```