Java Collections

Introduction to ArrayList and Hashmap

ArrayList

- Store a list of elements of any type

- Specify the **element type** with Java “Generics” syntax:
  - ArrayList**<Token>** tokenList = new ArrayList**<Token>**();

- ArrayList “wraps” Java’s built-in arrays
  - built in: int arr[]=new int[10];
  - Higher-level classes like ArrayList are safer to use

ArrayList methods

- add(<E> element) -- appends to end of list
- <E> get(int index) -- returns the indexth element
- int size()
- many more...

Sample ArrayList Code

```
Person joe= new Person();
joe.firstName="Joe";
joe.lastName="Jones";
Person zaza= new Person();
zaza.firstName="Zaza";
zaza.lastName="Gabor";

ArrayList<Person> persons= new ArrayList<Person>();
persons.add(joe);
persons.add(zaza);

// print all last names
for (int i=0;i<persons.size();i++) {
    Person p = persons.get(i);
    System.out.println(p.lastName);
}
```

Hashmap

- Map a String (typically) to some object.
- Like an ArrayList, but indexes are strings not integers
- Key-Value list

```
Hashmap<String,String> map = new Hashmap<String,String>();
map.put(“hola”,”hello”);
map.put(“adios”, “goodbye”);
```

Hashmap Methods

- put(key,value) -- adds a key-value pair
- get(key) -- returns value
- boolean containsKey(key) -- returns true if key in map
- many more...

Hashmap in Parser

SymbolTable “wraps” Hashmap

```
public class SymTab {

  HashMap<String,Integer> map = new HashMap<String,Integer>();

  // methods for adding to and getting things from SymTab.
```

Big Picture

- ArrayList and Hashmap part of Java Collections Framework

- Collection framework includes inheritance and interfaces, we’ll discuss these in detail later.
  - ArrayList and LinkedList implement List
  - HashMap is a subclass of AbstractMap which implements Map