THE UNIVERSITY OF ALABAMA AT BIRMINGHAM

Array and ArrayList

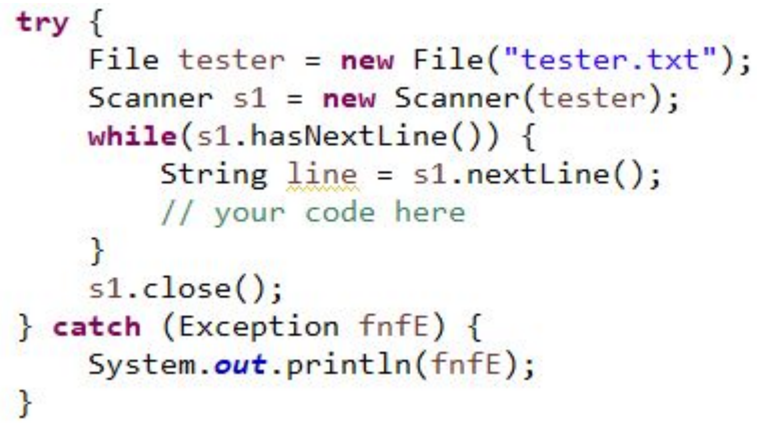
CS203 Lab 8

Concepts

- Read file
- Storing objects in an array or ArrayList
    - Store Dog objects in an array
    - Store Dog objects in an ArrayList
- Walk through elements in an array or ArrayList

Reading a file in Java

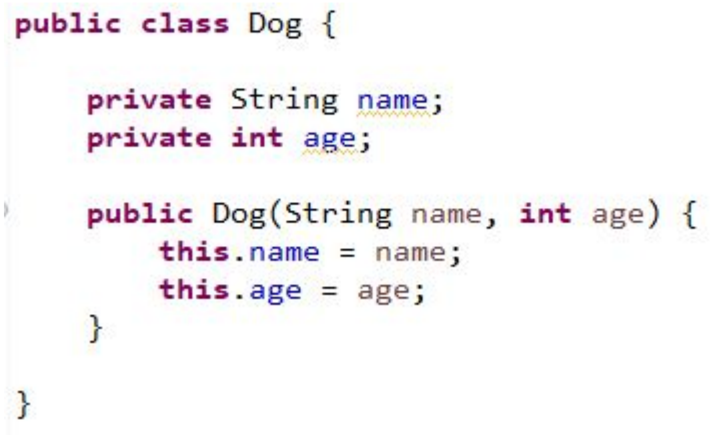
- Create a try catch statement
  - For the catch, add your (FileNotFoundException fnfE)
- Create a File object using the name of the input file
- Create a Scanner object using that File object
- Then, use methods from the Scanner class to read in data
  - myScanner.hasNextLine()
  - myScanner.nextLine()
- Make sure to close your file!



```
try {
    File tester = new File("tester.txt");
    Scanner s1 = new Scanner(tester);
    while(s1.hasNextLine()) {
        String line = s1.nextLine();
        // your code here
    }
    s1.close();
} catch (Exception fnfE) {
    System.out.println(fnfE);
}
```

For example’s sake, here’s a Dog class

Dog newDog = new Dog(“Dixie”, 8);



```
public class Dog {

    private String name;
    private int age;

    public Dog(String name, int age) {
        this.name = name;
        this.age = age;
    }

}
```

Storing objects in an array

- Define the variable type with square brackets and array name
  - int[] numArr;
  - Dog[] dogArr;
- Can initialize an array either way:
  - int[] numArr = {0,0,0,0,0,0,0,0,0,0}
  - Dog[] dogArr = new Dog[10]; **will have 10 null items in dogArr**
- Index positions in the array and reassign them new values
  - numArr[0] = 9;
  - numArr[2] = 7;
  - numArr[9] = 3;
  - This will return {9,0,7,0,0,0,0,0,0,3}
- Can do either:
  - Dog myNewDog = new Dog(“Dixie”,8);
  - dogArr[0] = myNewDog;

    or…
  - dogArr[0] = new Dog(“Dixie”,8);

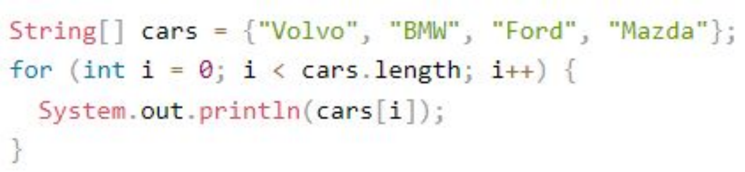
Storing objects in an ArrayList

- Import the ArrayList class from the util package from the Java Standard Library
    - import java.util.ArrayList
- Initialize the ArrayList
    - ArrayList<Dog> dogArr = new ArrayList<Dog>();
- Add items to the ArrayList
    - Dog myNewDog = new Dog(“Dixie”,8);
    - dogArr.add(myNewDog);

      or…
    - dogArr.add(new Dog(“Dixie”,8));
- dogArr.get(i); access an element at a certain index
- dogArr.set(i, anotherDog); modify/reassign an element at a certain index
- dogArr.remove(i); remove an element at a certain index
- dogArr.clear(); clear all elements in the ArrayList
- dogArr.size(); returns an int for how many elements there are
    - **the .size() method is VERY useful inside a for loop**

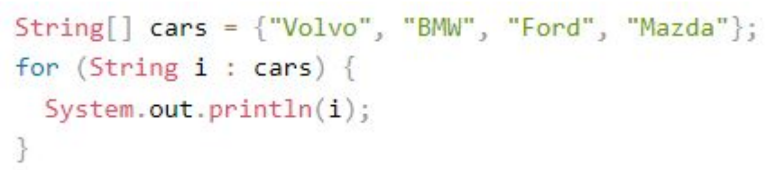
Walking through elements in an array

Use either a **for loop**



```
String[] cars = {"Volvo", "BMW", "Ford", "Mazda"};
for (int i = 0; i < cars.length; i++) {
  System.out.println(cars[i]);
}
```

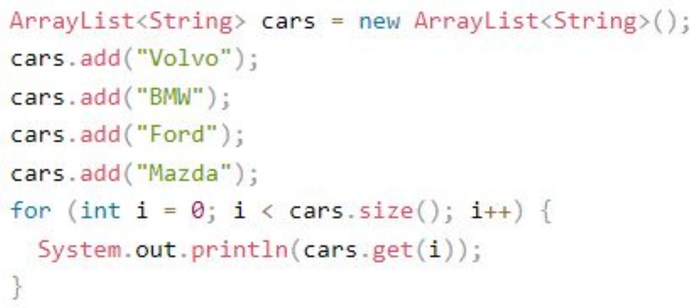
or a **for each loop**



```
String[] cars = {"Volvo", "BMW", "Ford", "Mazda"};
for (String i : cars) {
  System.out.println(i);
}
```

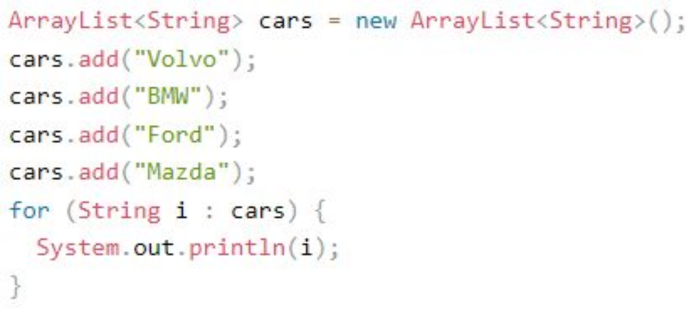
Walking through elements in an ArrayList

Use either a **for loop**



```
ArrayList<String> cars = new ArrayList<String>();
cars.add("Volvo");
cars.add("BMW");
cars.add("Ford");
cars.add("Mazda");
for (int i = 0; i < cars.size(); i++) {
  System.out.println(cars.get(i));
}
```

or a **for each loop**



```
ArrayList<String> cars = new ArrayList<String>();
cars.add("Volvo");
cars.add("BMW");
cars.add("Ford");
cars.add("Mazda");
for (String i : cars) {
  System.out.println(i);
}
```

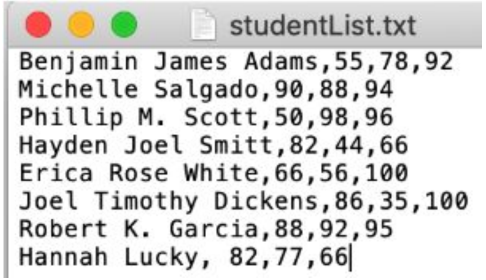
Lab 8 Assignment

```
public class Student
{
        private String Name;
        private Sting lastName;
        private int exam1,exam2, finalExam;
        private double finalGrade;
        private String letterGrade;
        public Student()
        {
                // create your constructor
        }

// create all of the necessary accessor and mutator methods

// Create the necessary methods to compute the letterGrade


}
```


studentList.txt

```
Benjamin James Adams,55,78,92
Michelle Salgado,90,88,94
Phillip M. Scott,50,98,96
Hayden Joel Smitt,82,44,66
Erica Rose White,66,56,100
Joel Timothy Dickens,86,35,100
Robert K. Garcia,88,92,95
Hannah Lucky, 82,77,66
```

Grade: Complete the Student.java file → 25 points

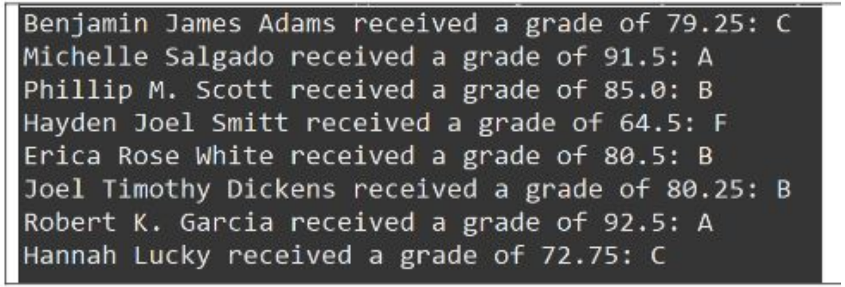
Read from file → 25 points

Store the values in array → 25 points

Compute and display the results → 25 points

Print out the result into a txt file → +20 points

The output can look something like this:



```
Benjamin James Adams received a grade of 79.25: C
Michelle Salgado received a grade of 91.5: A
Phillip M. Scott received a grade of 85.0: B
Hayden Joel Smitt received a grade of 64.5: F
Erica Rose White received a grade of 80.5: B
Joel Timothy Dickens received a grade of 80.25: B
Robert K. Garcia received a grade of 92.5: A
Hannah Lucky received a grade of 72.75: C
```