ArrayList<String>items=new ArrayList<String>();

Overloaded methods: you can have more than one method that has the same method but when you call the method java will know which one to call because there will be different parameters/arguments in the ()/constructor. It will automatically start with 10 objects but will create a new one that is larger. Or you can create an overloaded constructor (30) to prevent it from creating a new array list.

Items.add(“red”); to add an item to list

Items.add(0, “yellow”); (you can only put it there if there is already something there)

System.out.println(items.size()); it would be zero if there is no data.

Items.get(3) is the same thing is names[] for example

System.out.printf(“%”, items.get(i0))

For (String item: items)

Items.remove(“red”) it will remove the first occurance

Or items.remove(0)

Val= x>y ? x:y; it is conditional. Is x>y true? If it is then val=y if its not then val=y

System.our.printf(“\”red\” is %sin the list %n, Items.contains (“red”)?””: “not”) %s means replacing it with a variable type string

Item.remove(10);

System.out.println(items); is the same thing as System.println(items.toString()); but with brackets and commas. You can do a for loop to display it how you want

System.out.println(“The location of Denver in the list?” + cityList.indexOf(“Denver”));

System.out.println(“Is the list empty?” + cityList.isEmpty()); \\ print true of false

isEmpty(); returns True and False

for (int i=0; i<5; i++) {

double=imput.nextDouble();

list.add(x);

}

Get the total

For (int i=0; i<list.size()s; i++

Total+=list.get(i);

Or as in enhanced for loop

For( double val:list)

Total+= val;

Items.List.set(1, “hello”); this replaces the value