Java for Beginners

Level 6b

Mr. Teasdale

Levels of Java coding

- 1: Syntax, laws, variables, output
- 2: Input, calculations, String manipulation
- 3: Selection (IF-ELSE)
- 4: Iteration/Loops (FOR/WHILE)
- 5: Complex algorithms
- **6: Arrays/Linked Lists**
- 7: File management
- 8: Methods
- 9: Objects and classes
- 10: Graphical user interface elements

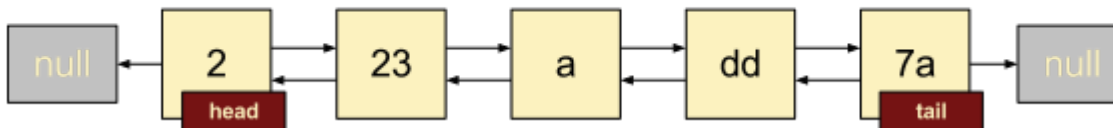# **Arrays** **vs** **Linked Lists**

Array

2	23	a	dd	7a
0	1	2	3	4

Array

Fixed size

One or Two dimensions

List

Size can change

Only linear

Linked List

null ← 2 ↔ 23 → a ↔ dd ↔ 7a → null

head   tail
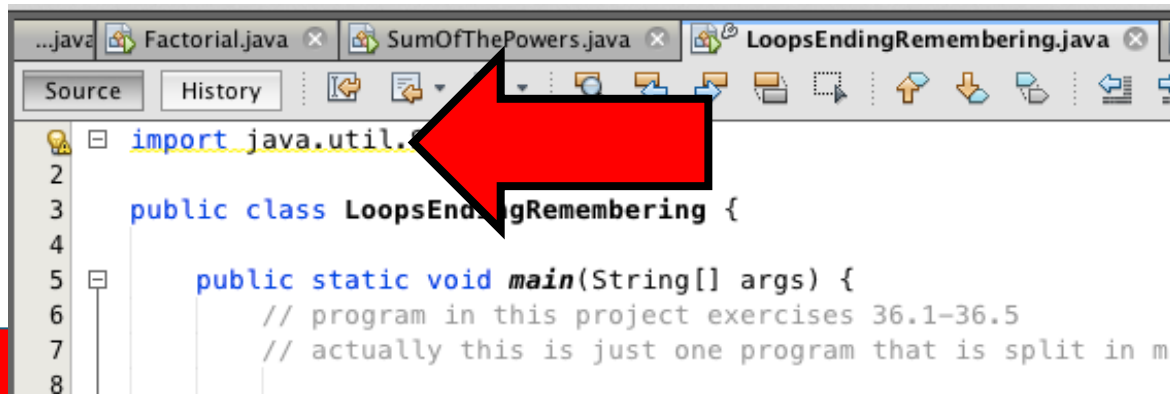
Before you using a Linked List…
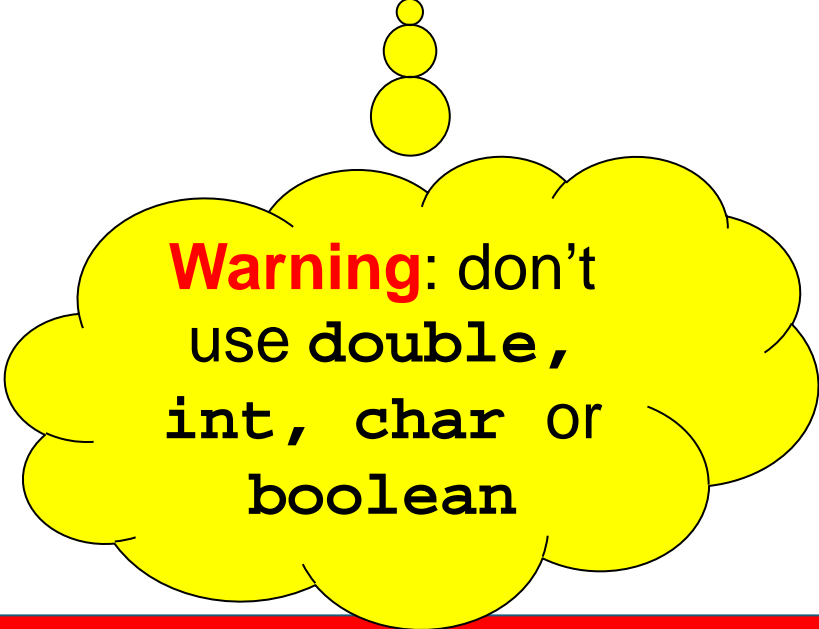
Remember to import the utility libraries!

`import java.util.*;`

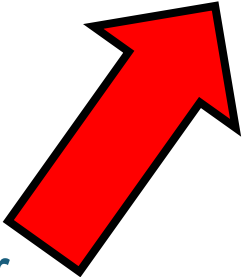# Creating a Linked List

```java
LinkedList<Integer> zones = new LinkedList<Integer>();

LinkedList<String> names = new LinkedList<String>();

LinkedList<Double> lines = new LinkedList<Double>();
```
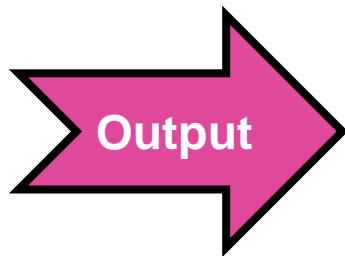
Warning: don't use **double, int, char** or **boolean**

Remember the **( )** at the end!
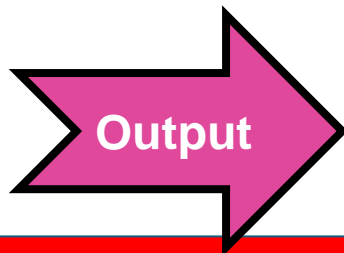
Adding items to a linked list

```java
LinkedList<String> names = new LinkedList<String>();

names.add("John");
names.add("Sam");
names.add("Chris");

System.out.println( names.get(0) );
```

Output → John

Removing items from a linked list

```
LinkedList<Integer> spots = new LinkedList<Integer>();

spots.add(23);
spots.add(15);
spots.add(69);

spots.remove(1);

for (int i = 0; i < spots.size(); i++)
{
    System.out.println( spots.get(i) );
}
```

Output →

```
23
69
```

Useful LinkedList methods

Method	What does it do
`.add(xx)`	Adds xx onto the end of the linked list
`.remove(y)`	Removes the element at position y
`.size()`	Returns how many elements there are in the linked list
`.indexOf(xx)`	Returns what element xx is stored in; returns -1 if element was not found

Beware getting the size!

`.size()` → Linked Lists

e.g. `int k = zones.size();` //zones is a linked list

`.length()` → Strings

e.g. `int m = name.length();` //name is a String

`.length` → arrays

e.g. `int g = boxes.length;` //boxes is an array

LinkedList example

```java
// create a linked list
LinkedList Letters = new LinkedList();
// add elements to the linked list

Letters.add("u");
Letters.add("g");
Letters.add("s");
Letters.add("b");
Letters.add("u");
Letters.add("r");

System.out.println("Content so far: " + Letters);
System.out.println("Oops, something is wrong here ... adding");
Letters.addLast("g");
Letters.addFirst("A");

System.out.println("Corrected contents of Letters: " + Letters);

// remove elements from the linked list
Letters.remove("s");
Letters.remove("g");
System.out.println("Contents of Letters after deleting 's' and 'g': "
+ Letters);
// remove first and last elements
Letters.removeFirst();
Letters.removeLast();
System.out.println("Letters after deleting first and last: "
+ Letters);
```

What did the programmer forget to do…?