# ArrayList

## COMP 102

Victoria University of Wellington

# Storing lots of values

- In most programs you need to remember *lots* of values:
  - all the lollies on the screen,
    so that you move them around or eat them later
  - all the numbers from a file,
    so that you can sort them all
  - all the names and marks of students in a class,
    so you can update the marks and display them
- Impractical to use lots of variables:

  Lollipop f1, f2, f3, f4, f5, f6, f7, f8, f9, … f100;

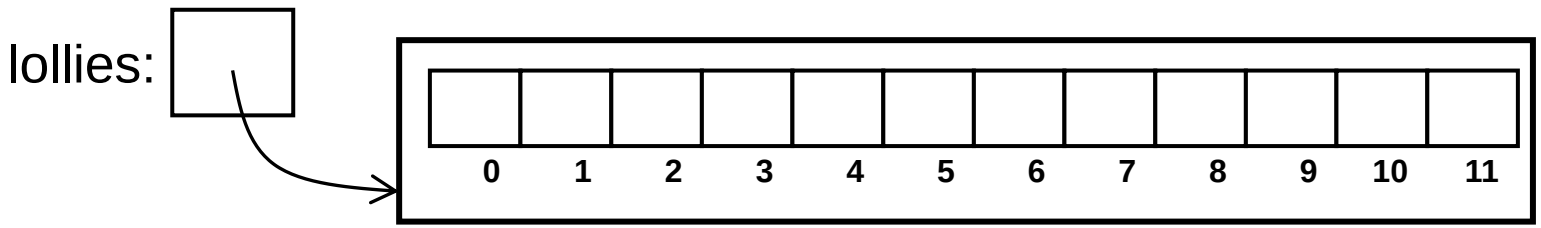
  int n1, n2, n3, n4, n5, … n1000;

  (The code would be huge!)
- Sometimes the number of needed variable is unknown when you make the program.
  - When we write the code we do not know how many students there will be in each class
  - Students might drop out!

# Arrays and ArrayLists

- Java has several structures that will hold lots of values:

Arrays and ArrayLists:




  - The individual places are numbered (from 0)
  - Can refer to the places by the index

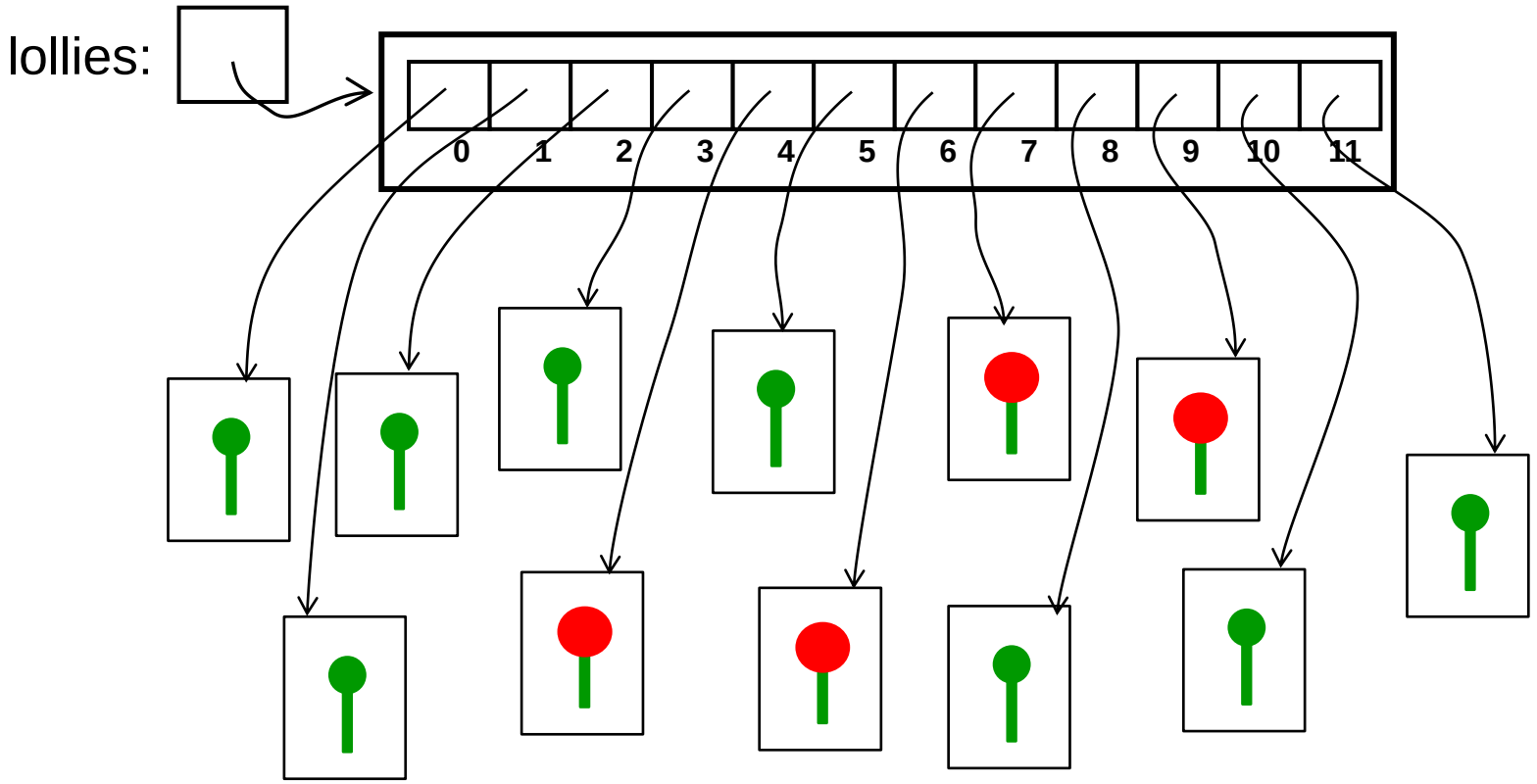
- Arrays: Fixed size, operations using special syntax
- ArrayLists: Grows as needed, all operations by method calls

# ArrayList of Lollipops

- lollies variable contains an ArrayList of Lollipop objects,

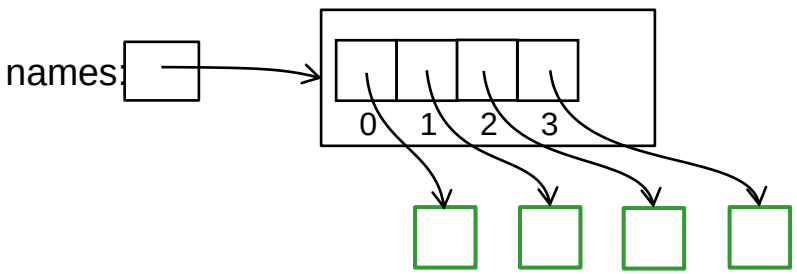
lollies.get(5) is the Lollipop at position 5

lollies.get(i).bloom() calls bloom on the Lollipop at position i.
(i must be less than the size of the ArrayList)



# Using ArrayList

- How do we
  - create a new ArrayList?
  - add a new value to the ArrayList?
  - access the values in the ArrayList?
  - change the value at an index in the ArrayList?
  - remove a value from the ArrayList?
  - do something to every value in the the ArrayList
  - find out how big the ArrayList is?



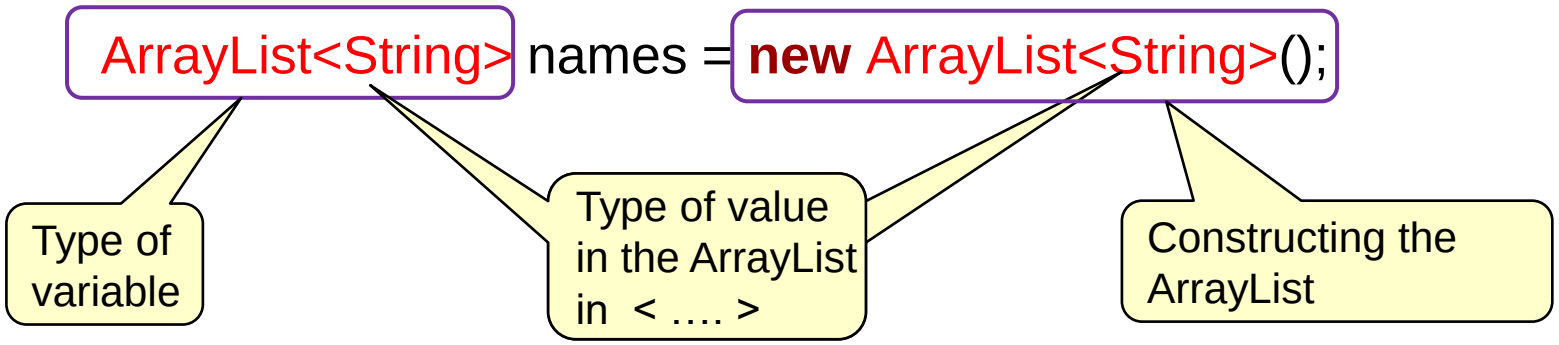
# Import java.util.*

- We need to import the standard java util library

```
import ecs100.*;
import java.util.*;
```

# Using ArrayList

- Creating an ArrayList object to store strings an assign it to a variable:




  - Must specify the type of value
  - Can't be int or double or boolean!!!
    Have to use Integer or Double or Boolean ("wrapper" objects)

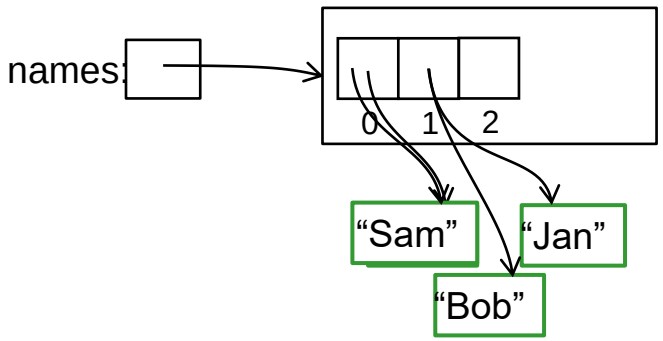
    ```
    ArrayList<Integer> counts = new ArrayList<Integer>();
    ```

  - Must have the ( ) - standard constructor, with no arguments.

# Using ArrayList

- For all actions, call methods on the ArrayList:
  - size, add, get, set, remove, clear, contains, indexOf, isEmpty, …

```
ArrayList<String> names = new ArrayList<String>();

names.add("Jim");

names.add("Jan");

names.add(1, "Bob");

names.remove("Bob");

names.set(0, "Sam");
```



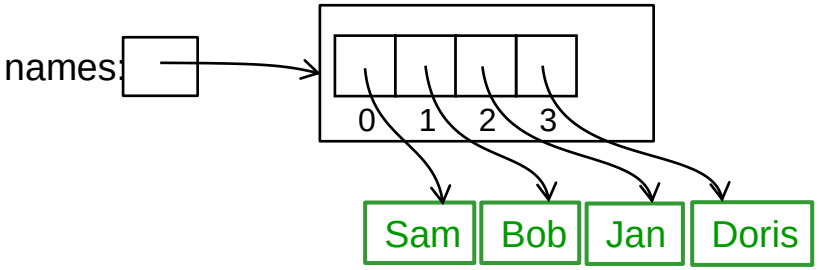
# Using an ArrayList: loops

- Usually, want to act on lots of elements

  ⇒ access and assign ArrayList elements inside a loop:

  - Print out all the names:

```
int i= 0;
while (i < names.size()) {
    UI.println( "Hello " ,  names.get(i));
    i = i +1;
}
```




  - Change each name to uppercase:

```
int num= 0;
while (num < names.size()) {
    String upName = names.get(num).toUpperCase();
    names.set(num, upName);
    num++;
}
```