

---

# Using arrays

## COMP 102

Victoria University of Wellington

# Using an array

- Can act on the whole array (like ArrayList)

- to pass to a method
- to assign to another variable

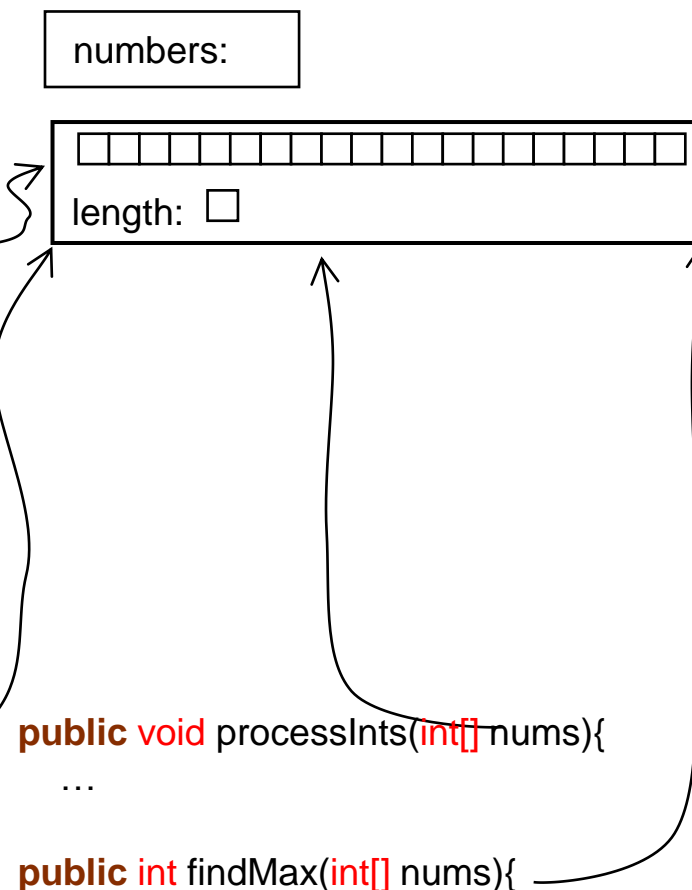
```
int [ ] numbers = new int[20];
```

```
//code to fill the array
```

```
this.processInts(numbers);
```

```
int maxNum = this.findMax(numbers);
```

```
int [ ] windowSizes = numbers;
```



- Note, passing as argument and assignment do **not** copy the array! (just the reference/ID of the object)
- Just the same as with ArrayList.

# Using an Array

- Use [ .. ] to refer to an individual place in the array
  - to access the value in that place
  - to put a value in that place (using assignment: = )

Not get() and set()

```
double [ ] marks = new double [200];
```

```
int n=4;
:
```

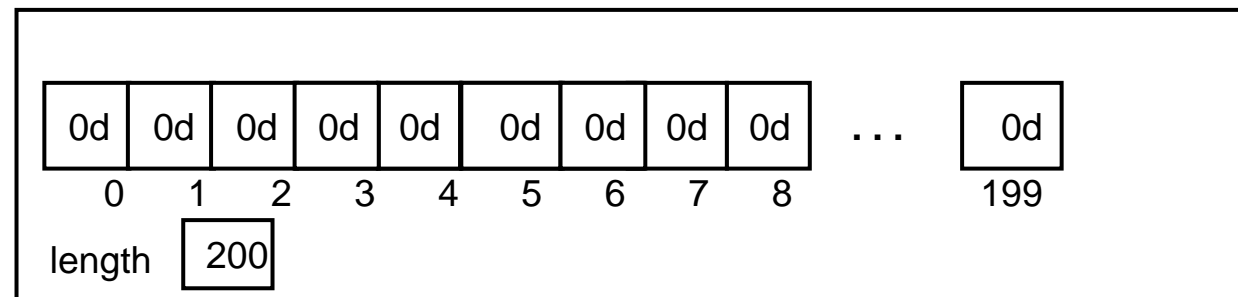
```
marks[5] = 45.6;
```

```
marks[6] = ( marks[5] + marks[7] ) / 2;
```

```
marks[n-1] = 80.0;
```

```
marks[n] = marks[n+1];
```

```
if (marks[ i ] == marks[ i+1 ]) {...
```



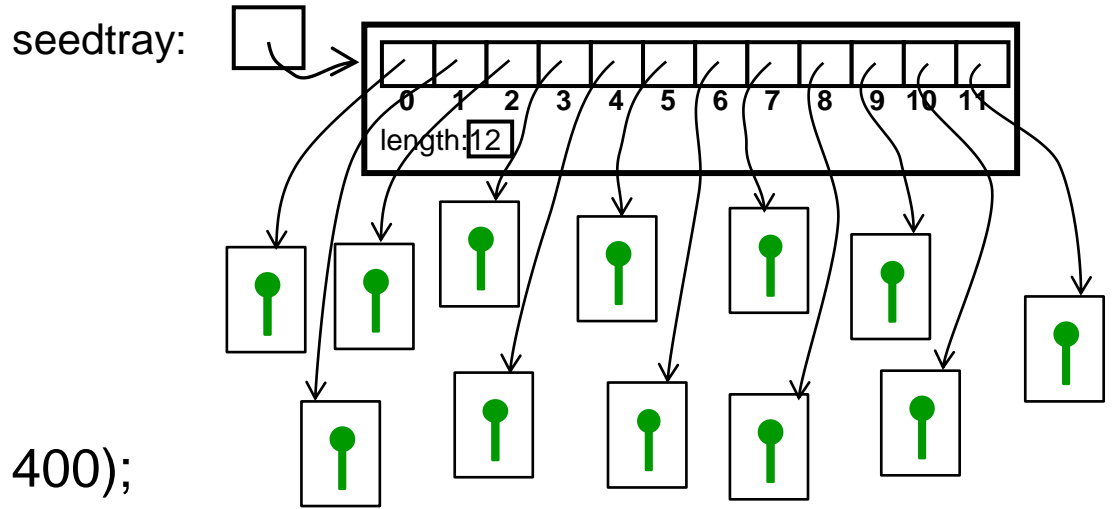
Index can be any **int** valued expression

# SeedTray Program

```

public class SeedTray{
    private Flower[] seedtray = new Flower[12];
    :
    public void replant(){
        for (int i = 0; i < this.seedtray.length; i++) {
            this.seedtray[ i ] = new Flower(70+i*50, 400);
        }
    }
    public void growAll(){
        for (int i = 0; i < this.seedtray.length; i++) {
            this.seedtray[ i ].grow();
        }
    }
}

```



```

public void growAll(){
    for (Flower flower : this.seedtray){
        flower.grow();
    }
}

```

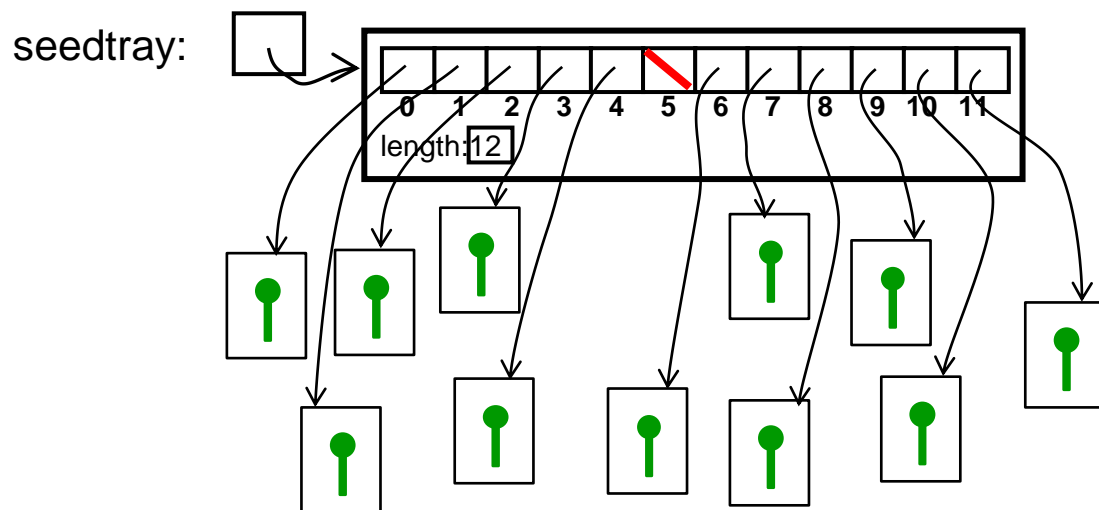
For each loop works on arrays, just like ArrayLists

# Arrays of Objects can contain null

```
public void pick(int index){
    this.seedtray[ index ] = null;
}
```

If the array may have null, must check items before acting on them

```
public void growAll(){
    for (int i = 0; i < this.seedtray.length; i++) {
        if (this.seedtray[ i ] != null){
            this.seedtray[ i ].grow();
        }
    }
}
```

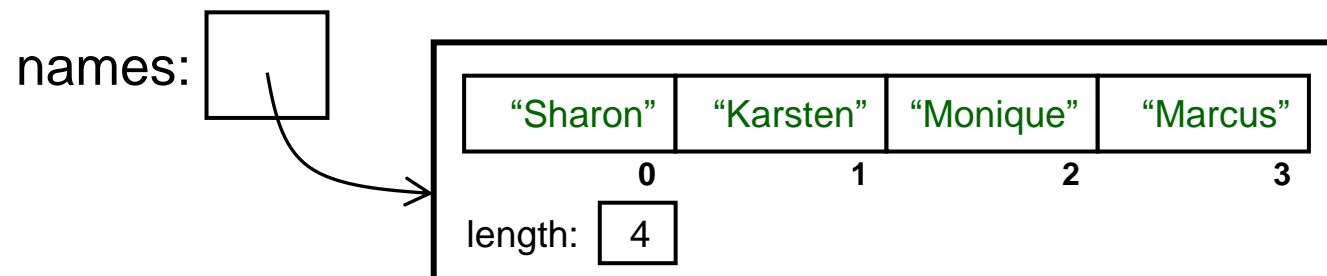


```
public void growAll(){
    for (Flower flower : this.seedtray){
        if (flower != null){
            flower.grow();
        }
    }
}
```

# Initialising the contents of an array

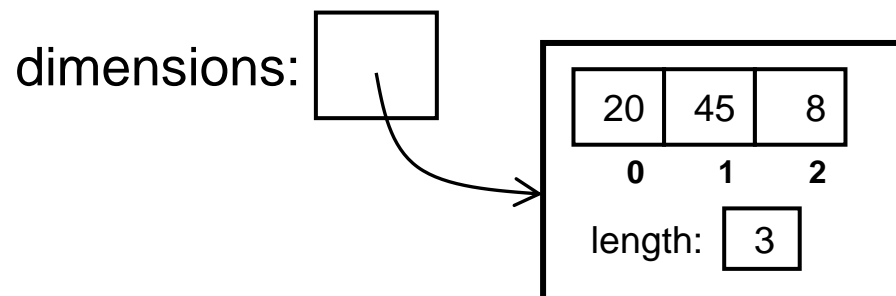
- Can specify the initial values (and size) of an array by listing the values in `{... , ... , ...}` :

```
String [] names = new String [] { "Sharon", "Karsten", "Monique", "Marcus" };
```



Can't do this  
with ArrayLists!

```
int [] dimensions = new int [] { 20, 45, 8 };
```



# Arrays vs ArrayList

---

- Use an array if
  - it will never change size, and
  - you know how big it will need to be, at the point you need to create it.
- Use an ArrayList if
  - the size will change, or
  - you don't know how big it will need to be.
- Arrays have convenient syntax `[]`
- ArrayLists have convenient methods.