[TAP:WJOGN] Casting

```
Integer i = (Integer) 10; //1
Baby baby1 = (Baby) new BossBaby("Bill",3); //2
int rounded = (int) 1~~.8~~;//3
```

int

- Which of the above explicit castings is necessary?
  - A. 1
  - B. 2
  - [✓ C. 3]
  - D. 1 and 3
  - E. Whatever

Administrative Details

- Lab 2
  - Complete PRE-LAB before lab

Agenda

- Lab2
- Array
- Vector

Lab 2 Overview

- 1. Given an input text, build tables of letter frequencies:
  - For each String (of length 1, 2, 3, …)
    - For each letter that follows the given String
      - Count the # of occurrence

Vector<Association<Character, Integer>> ← FrequencyList

Vector<Association<String, FrequencyList>> ← Table

["a" : [('a', 1)
('b', 1)],
"b" : [('u', 1)
('l', 1)], ...] ← k

length = 1

Lab 2 Overview

WordGen

- 2. Generate random “sentences” based on:
  - 1 previous character: k=1

    "Shand tucthiney m?" le ollds mind Theybooure
    He, he s whit Pereg lenigabo Jodind alllld ashanthe ainofevids tre
    lin--p asto oun theanthadomoere
  - 2 previous characters: k=2

    "Yess been." for gothin, Tome oso; ing, in to
    weliss of an'te cle - armit. Papper a comeasione, and smomenty,
    fropeck hinticer, sid, a was Tom, be suck tied. He sis tred a
    youck to themen

.' -

Lab 2: Generating a Sentence

- Given k=1,2,3,…
- Start building the "sentence" sb ← new StringBuffer ( );
- sb = first k letters of the input file
- while length < 500
  - Add to sb a new letter based on k previous letters
    - Get FrequencyList associated with the String of length k
    - Select a random character using the FrequencyList (Pick a random letter weighted by frequency)
- Convert sb to String

Lab 2: Generating a Sentence

- Picking a random letter weighted by frequency

[('a', 5) ('x', 4)
('b', 1)]

total = 10

```
Random r = new Random();
int n = r.randInt(total); // returns 0 ... total-1
```

50% 10% 40%

$\underbrace{0, 1, 2, 3, 4}_{\text{'a'}}, \underbrace{5}_{\text{'b'}}, \underbrace{6, 7, 8, 9}_{\text{'x'}}$

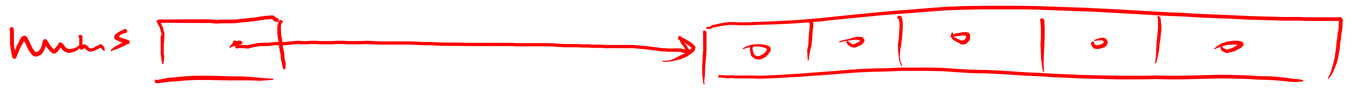
Agenda

- Lab2
- ⦿ Array
- Vector

Array

- An array is stored in consecutive memory locations:

```
int[] nums;
nums = new int[5];
```



nums[0]

nums[2]

mem location = mem location of Array + (index × size of type)

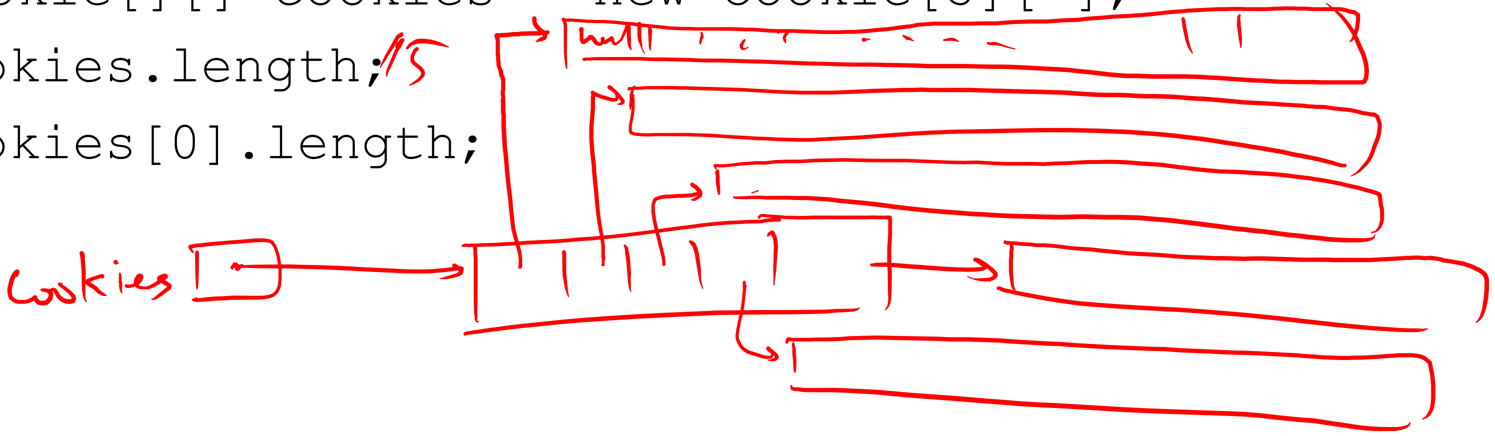
~~Multi-Dimensional~~ arrays of Arrays

- Syntax for 1-D array:

```
Cookie[] cookies = new Cookie[5];
cookies.length; //5
```

- Syntax for 2-D array:

```
Cookie[][] cookies = new Cookie[5][13];
cookies.length; //5
cookies[0].length; // 13
Cookie[][] cookies = new Cookie[5][ ];
cookies.length; //5
cookies[0].length;
```



Issues with Arrays

- What if you run out of space?
  - Too bad, you’ll need to create a new (bigger) array and copy everything!

Agenda

- Lab2
- Array
- Vector

Vector: A Flexible Array

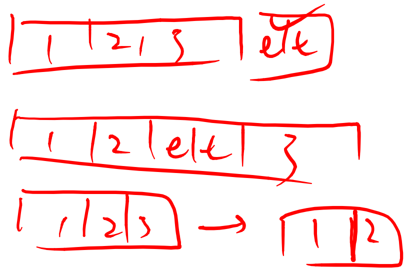
- Provides functionality of array
- Automatically “grows” as needed

```
public class Vector<E> {
        private ~~E~~ Object[] elementData;
        protected int elementCount;

                    ⋮

}
```

Vector Class : Methods



```
public void add(E elt)
public void add(int i, E elt)
public E remove(int i)
public int capacity()
public int size() // like "length" in array
public boolean isEmpty()
public E get(int i)
public E set(int i, E elt)
public boolean contains(E elt) { return indexOf(elt)==-1; }
public int indexOf(E elt) // returns -1 if not found
public void ensureCapacity(int minCapacity)
```

Extending the internal array

- How should we extend the array?
  - Grow by fixed amount when capacity is reached
  - Double array when capacity is reached

ensureCapacity

```
public void ensureCapacity(int minCapacity){
     if (elementData.length < minCapacity) {
          int newLength = elementData.length;
          if (capacityIncrement == 0) {
               // increment of 0 suggests doubling (default)
               if (newLength == 0) newLength = 1;
               while (newLength < minCapacity) {
                    newLength *= 2;
               }
          } else {// increment != 0 suggests incremental increase
               while (newLength < minCapacity) {
                    newLength += capacityIncrement;
               }
          }
          Object newElementData[] = new Object[newLength];
          for (int i = 0; i < elementCount; i++) {
               newElementData[i] = elementData[i];
          }
          elementData = newElementData;
     }
}
```

double the size

increment by fixed amount

copying element

determinin new length

WordFreq.java

- Goal: Count frequencies of each word in a file

Vector<Association<String, Integer>>

↑ ↑

word #