Grab

new Random().nextInt(count)

import java.util.Random

choose a random number

replace that element with ~~[illegible]~~ null
decrement count
return element

move last element to ~~[illegible]~~ replace the hole

What's the problem?

add won't work
frequency won't work

Remove

Solution: Take ~~[illegible]~~ element at count & put at index

find first example of elem

swap last
decrement count

return True

Going over what we aren't doing for a reason ~~[illegible]~~ As your problems get harder and you work with more people, you'll need to bring them along with you. You may go straight to a good solution but if you stop there, you may miss the best one. Also increasingly there is no right answer.

Song → Grading
Rubber Duck ~~[illegible]~~ Debugging
Bags
Grab / Remove

Sharing code, we can replace with a private method

Song
Private methods
Polymorphism
Casting
Object
Generics

What ultimately matters in this course ~~is where~~ you end up, ~~not where you start.~~

isn't where you end up relative to your classmates but relative to yourself in week 1

Bags → Flexible
Expandable
~~[illegible]~~
Linked Bag

More Java

private methods

- are not part of an interface
- are not available outside the class

- Often called "helper methods", which let you reuse code

Refactor
- change the implementation of an existing function without changing the public behavior

Polymorphism

- Literally : Many forms

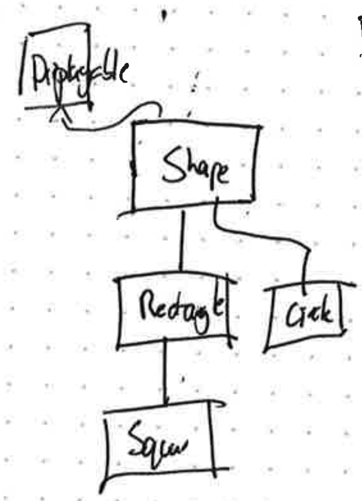
Single variable can hold many type of objects

Object Oriented ~~Paradigm~~ Programming

Abstraction
- Defining interfaces / ADT

Encapsulation
- Data Hiding

Inheritance
- "is -a" relationships
- sub / superclasses



Do NOT ERASE

DO NOT ERASE

Shape s = Rectangle
= Square()
= Circle()

Rectangle r = Rect
= Square

variable type — What is available

object type — what is evaluated

Casting

- ~~Java~~ is
- Must declare types in Java
- When there is a difference between variable type & reference type it is sometimes necessary to cast the variable to a new type

- Upcasting
  - Change up the class hierarchy
  - Java handles this automatically

- Downcasting
  - change down the class heirarchy
  - must do this explicitly
  - If the types are wrong, you get errors

Object Class

- Every class in Java inherits from the Object class (add to diagram)
- For any type ~~[illegible]~~ ~~(Other than primitives!~~)
    - Object o = new Integer(5)

~~- Has other methods~~

- Example methods on Object
    - String toString()
    - boolean equals(~~[illegible]~~ Object o)

Convert Bag to Objects

Still have to do explicit casting
- Mentally tracking types
- Likely causes errors

Generics

~~Type variables~~
~~Generic~~
Generic Types let you use a variety of types
- let you work with a Type before you know what it is
    - The data structure doesn't know ~~[illegible]~~ the type
    - The user of the data structure does

Advantages
- Avoids casting
- Language catches mistakes for us.

Convert to generic

Resizing a Bag

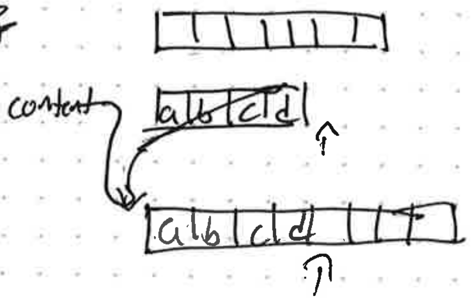
- We want to relax the restriction that a bag has fixed size
- How do we add now?

```
1. If bag is full return false
2. Else insert at count
3. Increment count
4. Return false
```

} called Psuedocode

```
1a If bag is full create a new array of size ~~[illegible]~~ count ~~[illegible]~~ 2
1b For each element of the original array, ~~[illegible]~~ copy it to the new array
1c Replace the old array with the new array
```

~~We will use Arrays.copyOf(array, new size)~~



Note Arrays.copyOf(   )

Time it

~~[illegible]~~ Big-O

How many steps does this take ~~[illegible]~~ in the worst case
- add : O(n)
- remove : O(n)
- grab : O(1)
- getFrequencyOf : O(n) → ~~[illegible]~~ make

~~Linked 1~~

Will talk more in depth on Big Oh

~~Linked Bag~~ We are asking for more than we need (but not very often), or we ask for to little.

Need to know madance

42

What if I just ask for 1 integer?

Ask for a new integer each time

What's the problem?

Don't know where the ints are

Solution? Store addresses