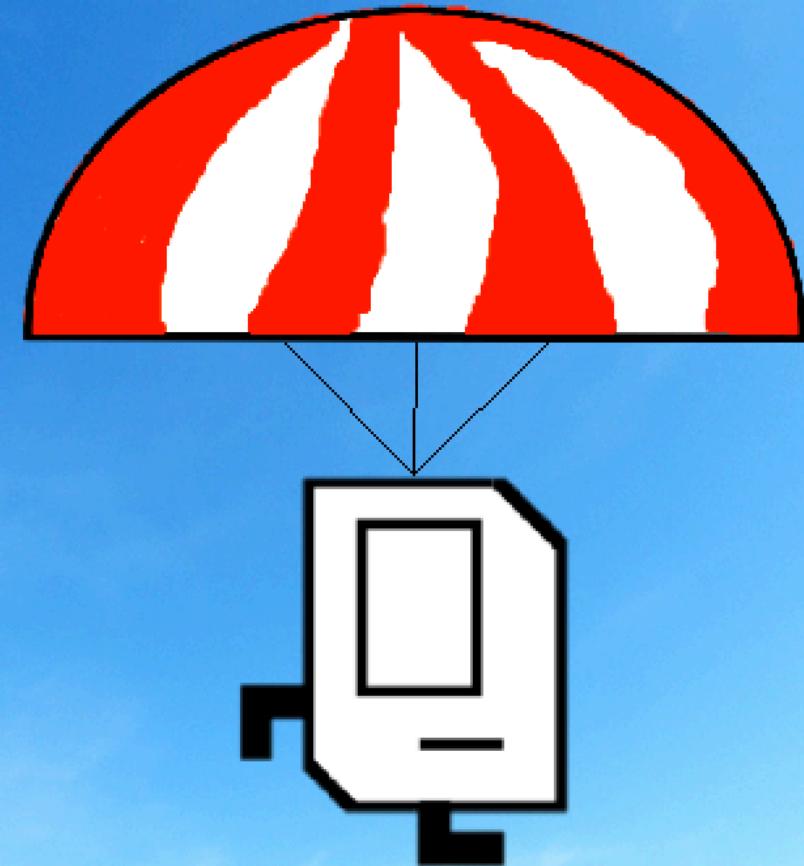


Hangman

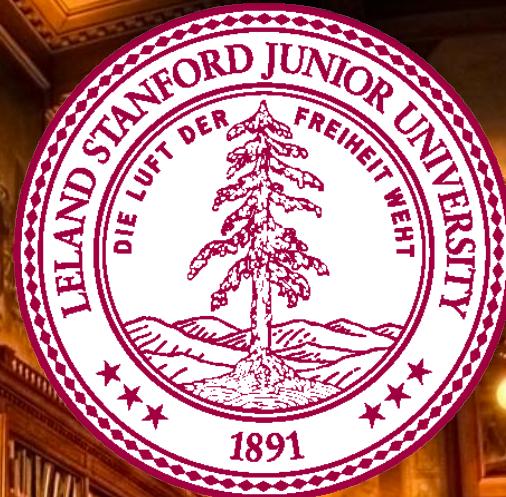
```
Welcome to Hangman
Your word looks like this: -----
You have 7 guesses left
Your guess: a
There are no A's in the word.
Your word looks like this: -----
You have 6 guesses left
Your guess: e
There are no E's in the word.
Your word looks like this: -----
You have 5 guesses left
Your guess: i
There are no I's in the word.
Your word looks like this: -----
You have 4 guesses left
Your guess: o
There are no O's in the word.
Your word looks like this: -----
You have 3 guesses left
Your guess: u
That guess is correct.
Your word looks like this: -U---
You have 3 guesses left
Your guess: z
That guess is correct.
Your word looks like this: -UZZ-
You have 3 guesses left
Your guess:
```



-UZZ-

AEIO

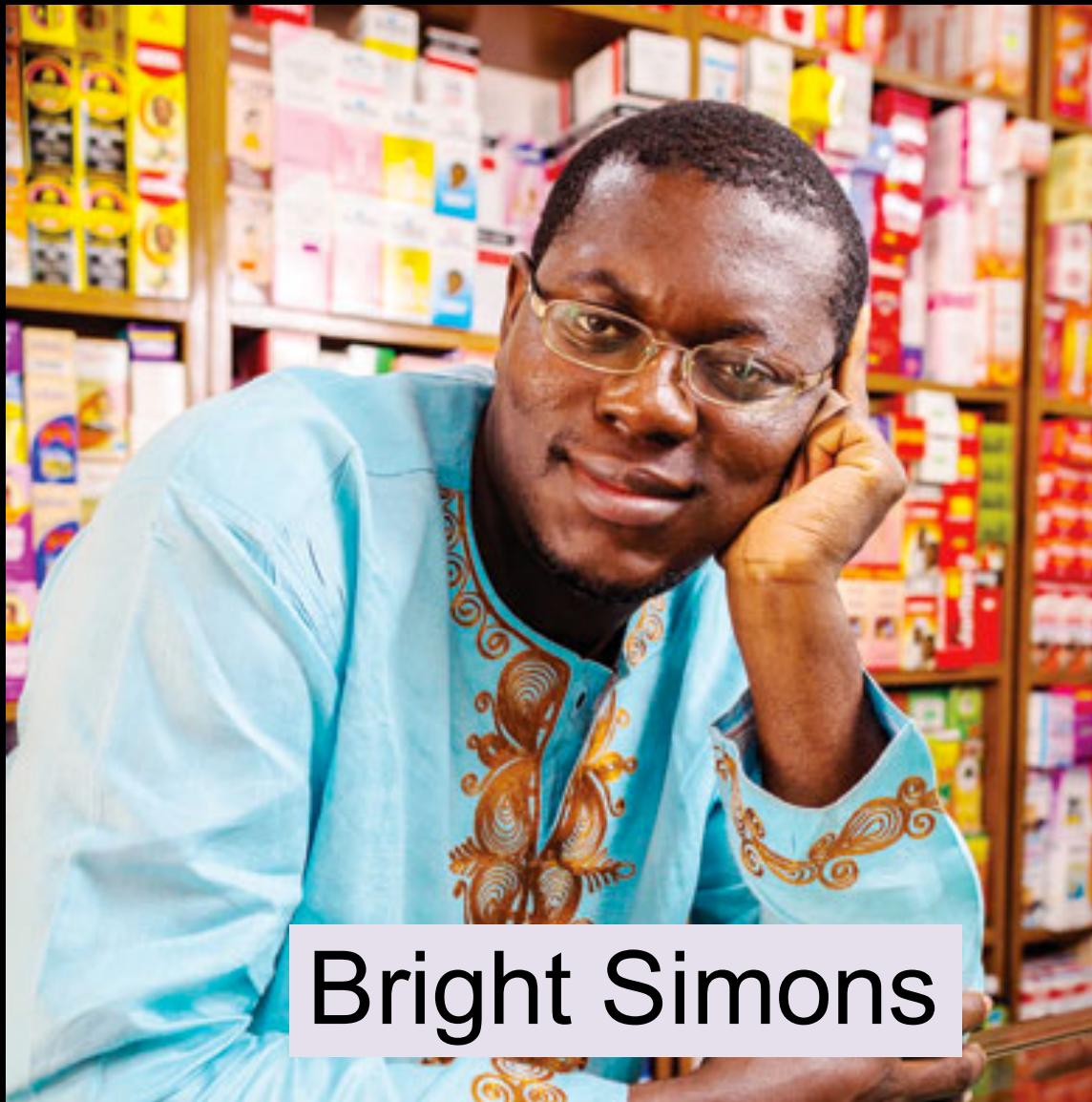




File Reading

Chris Piech
CS106A, Stanford University

Chris' Favorite Program



Bright Simons

Piech, CS106A, Stanford University



Underlying Puzzle

Counterfeiter



You (Distributor)

User

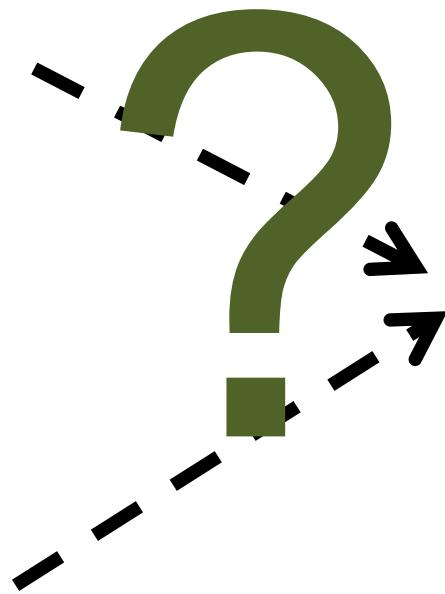


Underlying Puzzle

Counterfeiter



You (Distributor)



User



Make a code to
put on every box



1. Unique
2. Impossible to guess

Insight

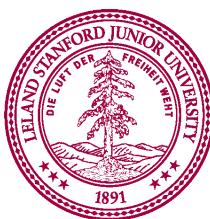
Code = RandomNum + UniqueNum

So that it is
impossible to guess

Concatenation

+

So that no two
codes are the same



M-Pedigree

MPedigree

4843220000
9861230001
2330240002
8047970003
1543690004
2787880005
9838840006
5224750007
2661390008
3482180009
4249170010
4133400011
1984670012
8917780013
6907970014
9829370015
3775510016
9956230017
0649500018
4208970019
1740950020
7023530021
9679450022



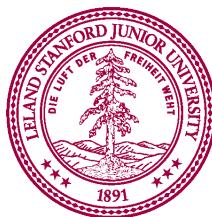
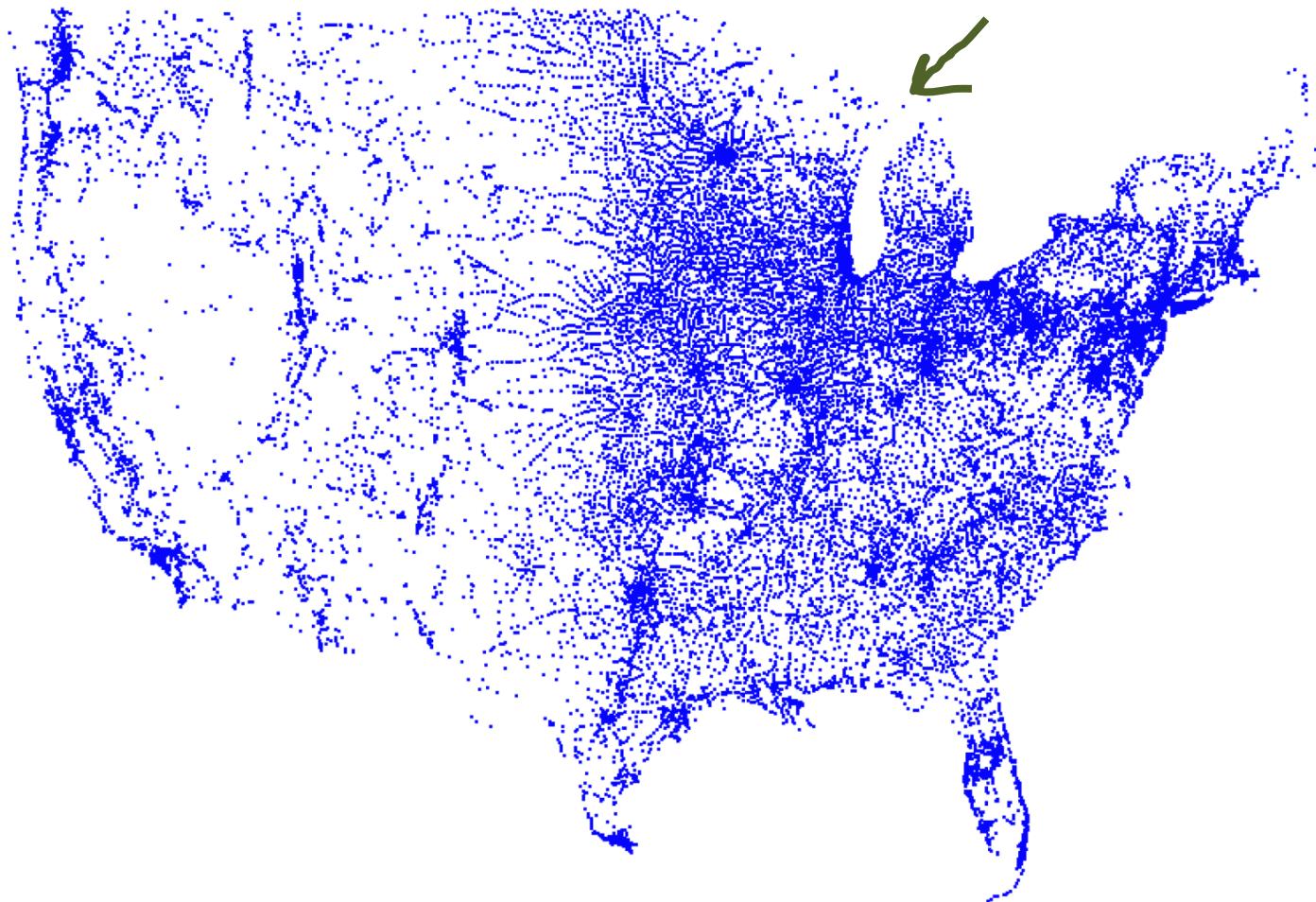
Learning Goals

1. Write string algorithms that loop over each character.
2. Know how to read a file line by line.





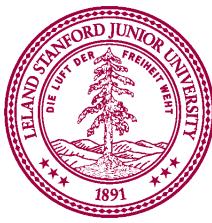
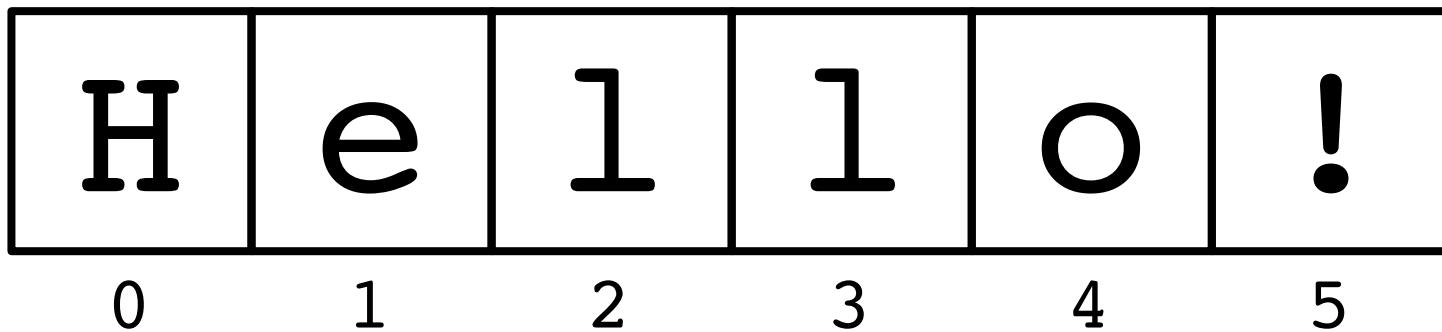
Each blue dot is a tiny
GRect



Review

How is text
represented?

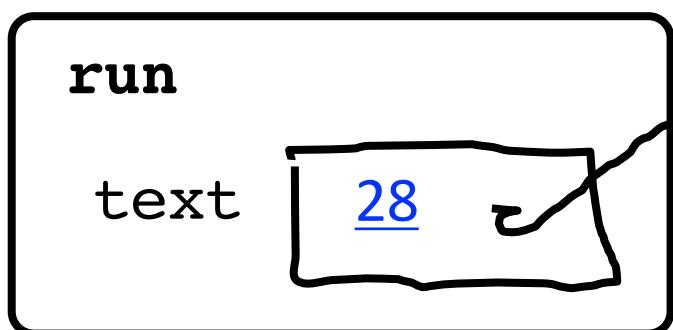
```
public void run() {  
    String text = "hello!";  
}
```



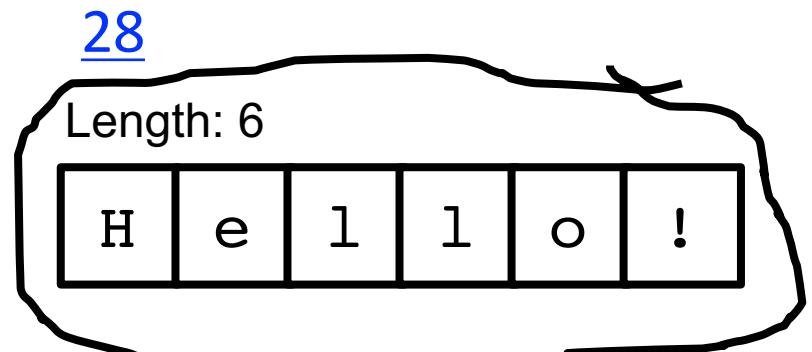
How it is actually stored

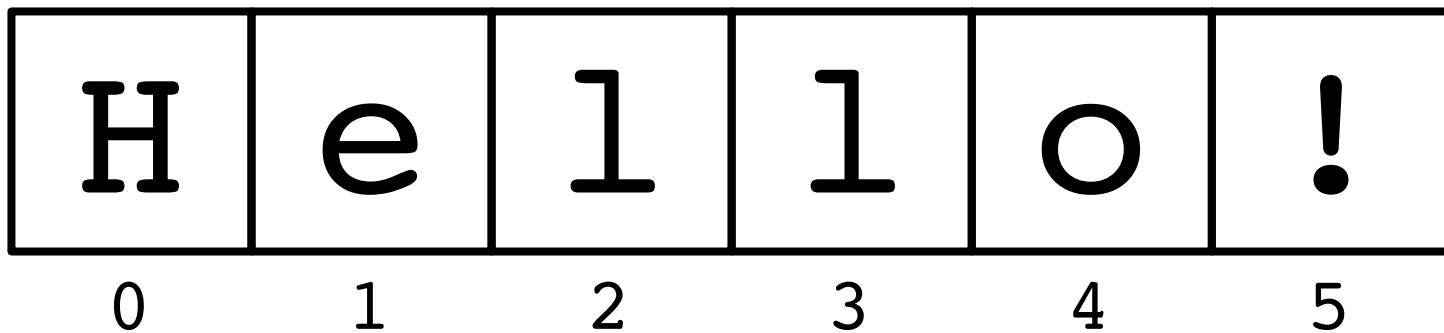
```
public void run() {  
    String text = "hello!";  
}
```

stack

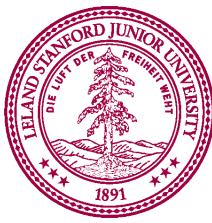


heap?





text.charAt(index)



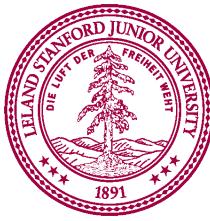


Piech, CS106A, Stanford University

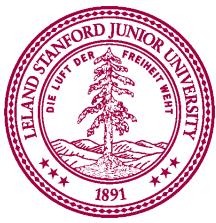
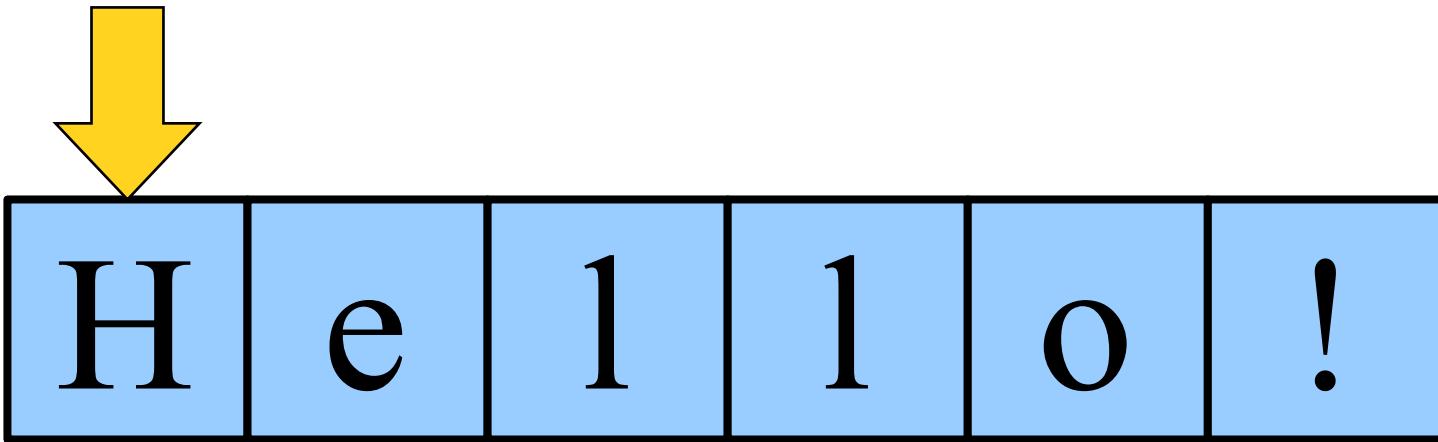




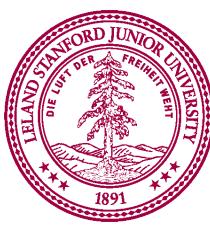
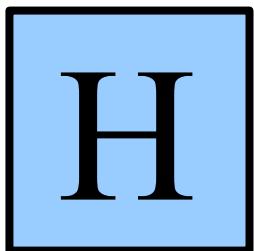
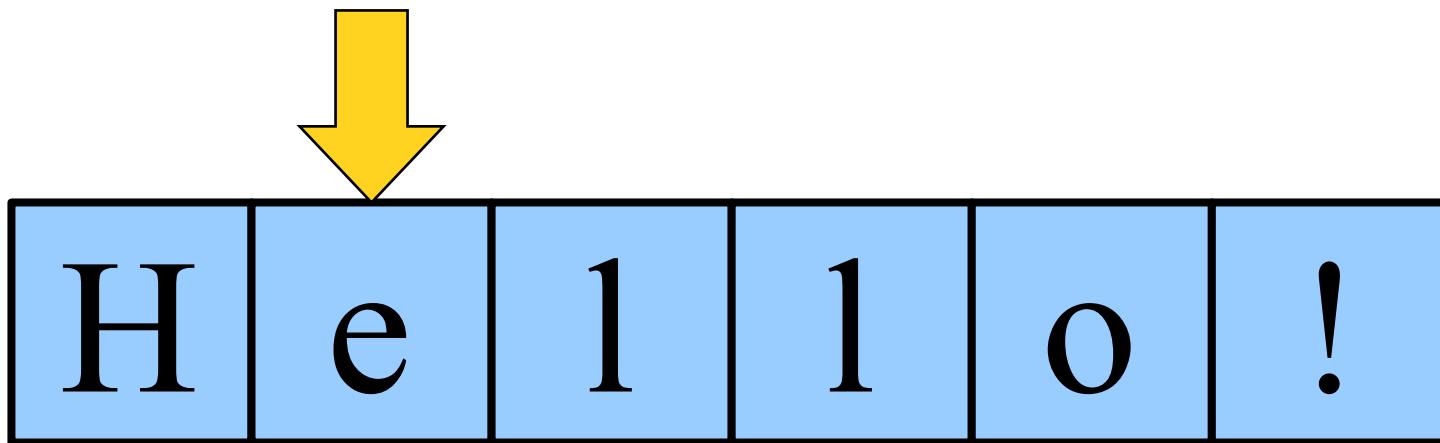
Many string algorithms use
the “loop and construct”
pattern.



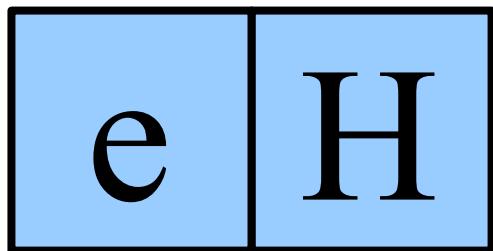
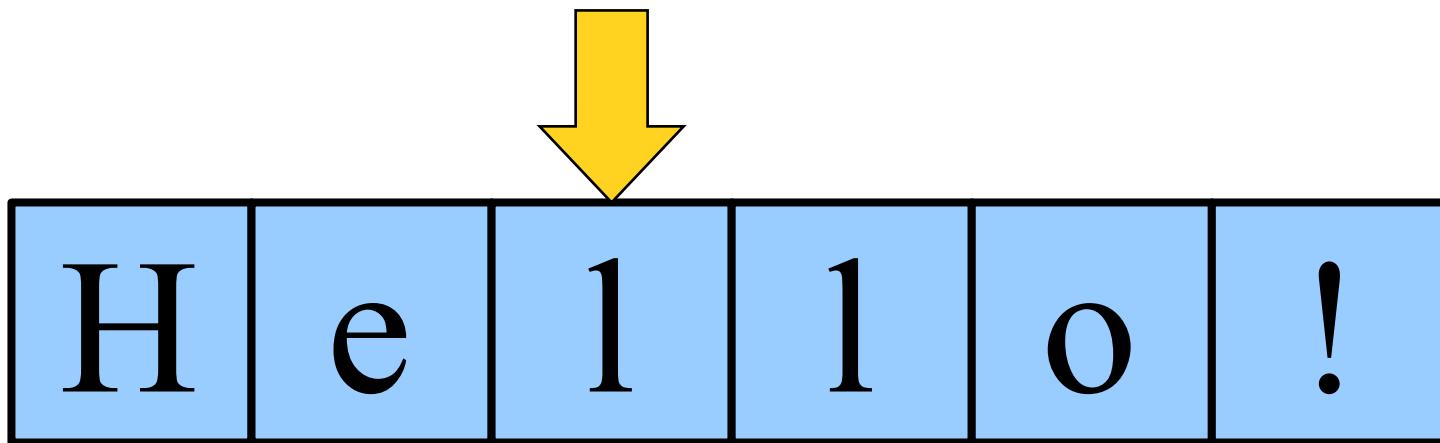
Reversing a String



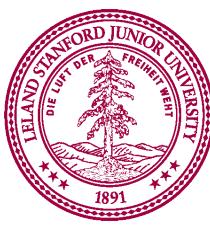
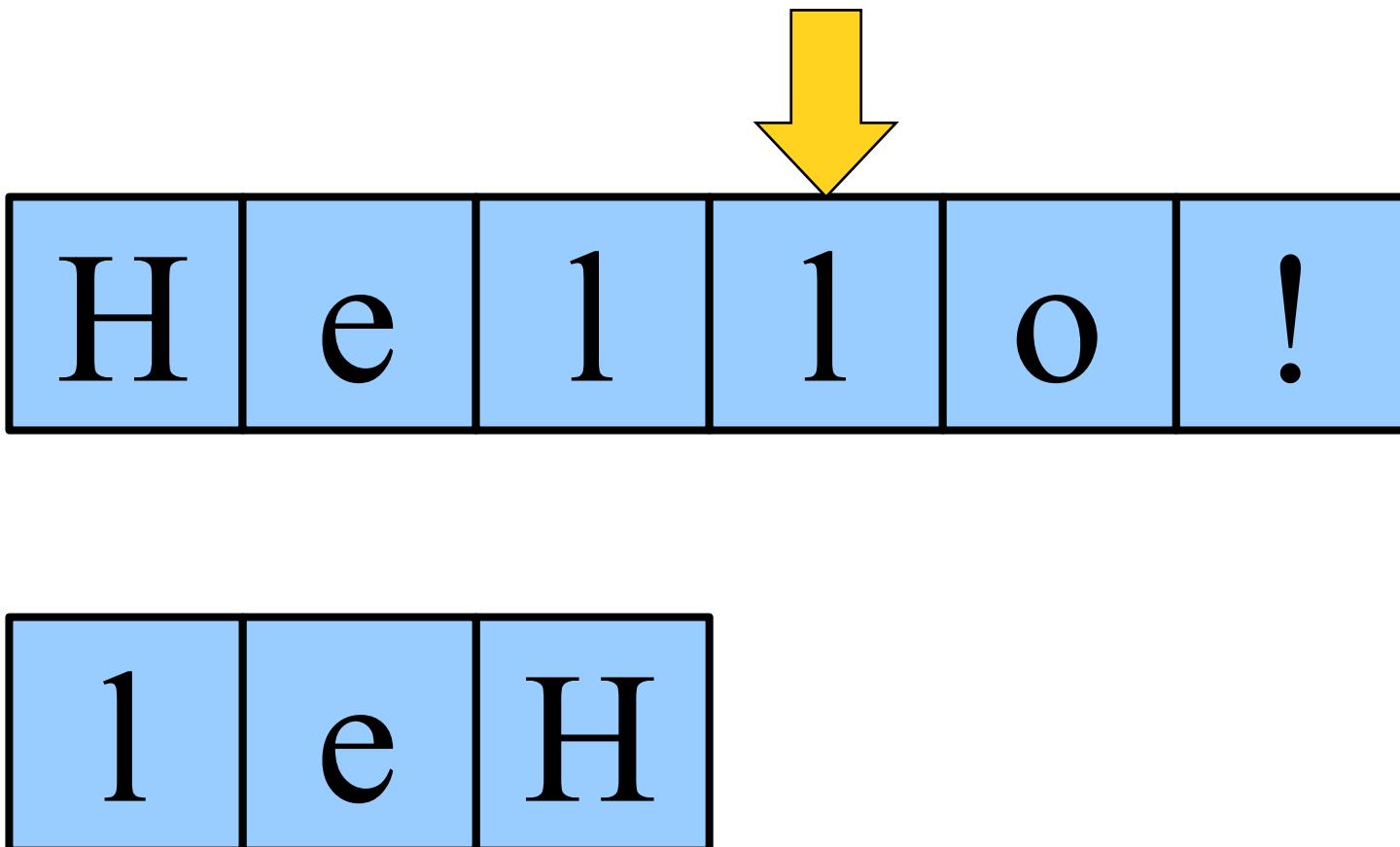
Reversing a String



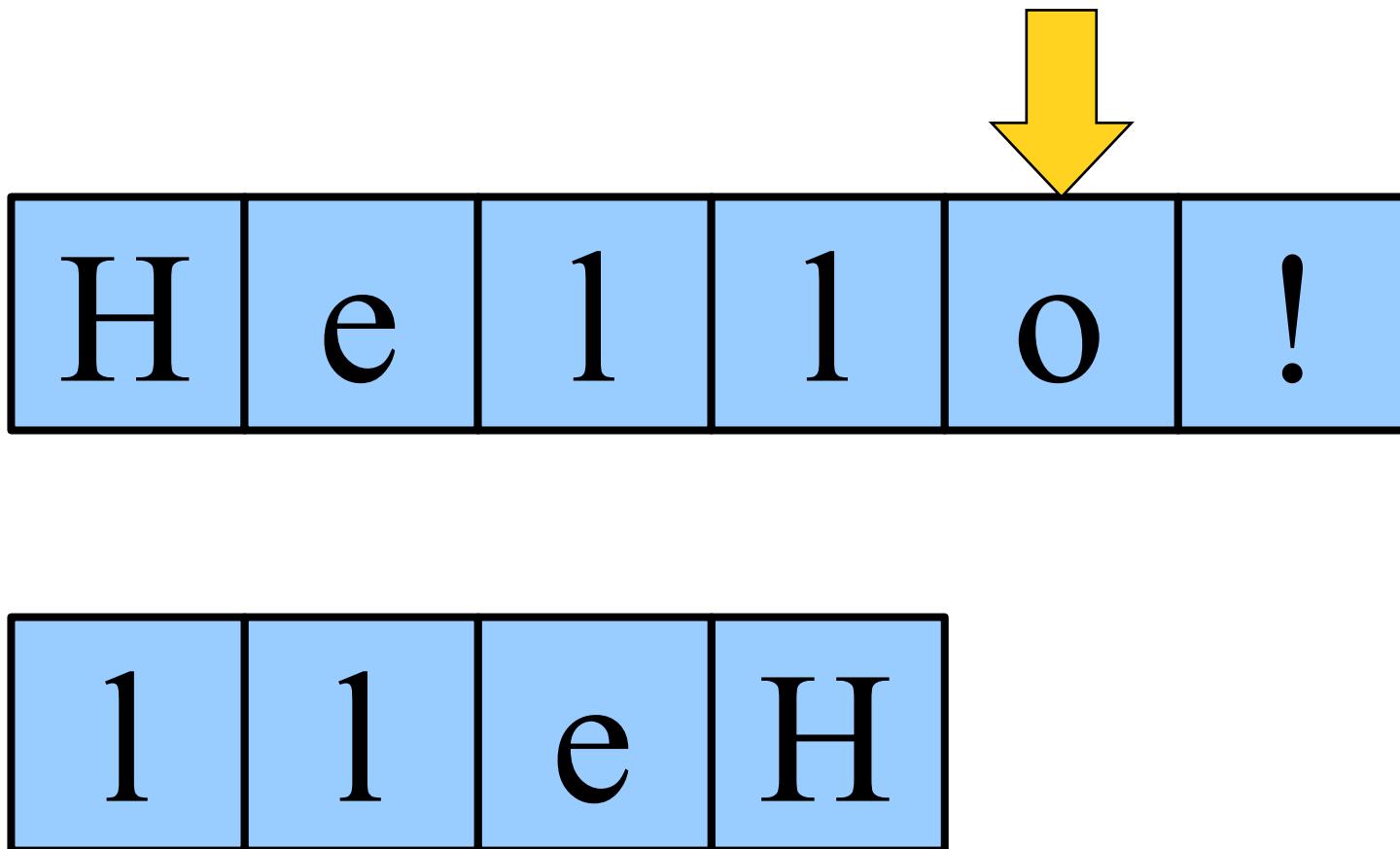
Reversing a String



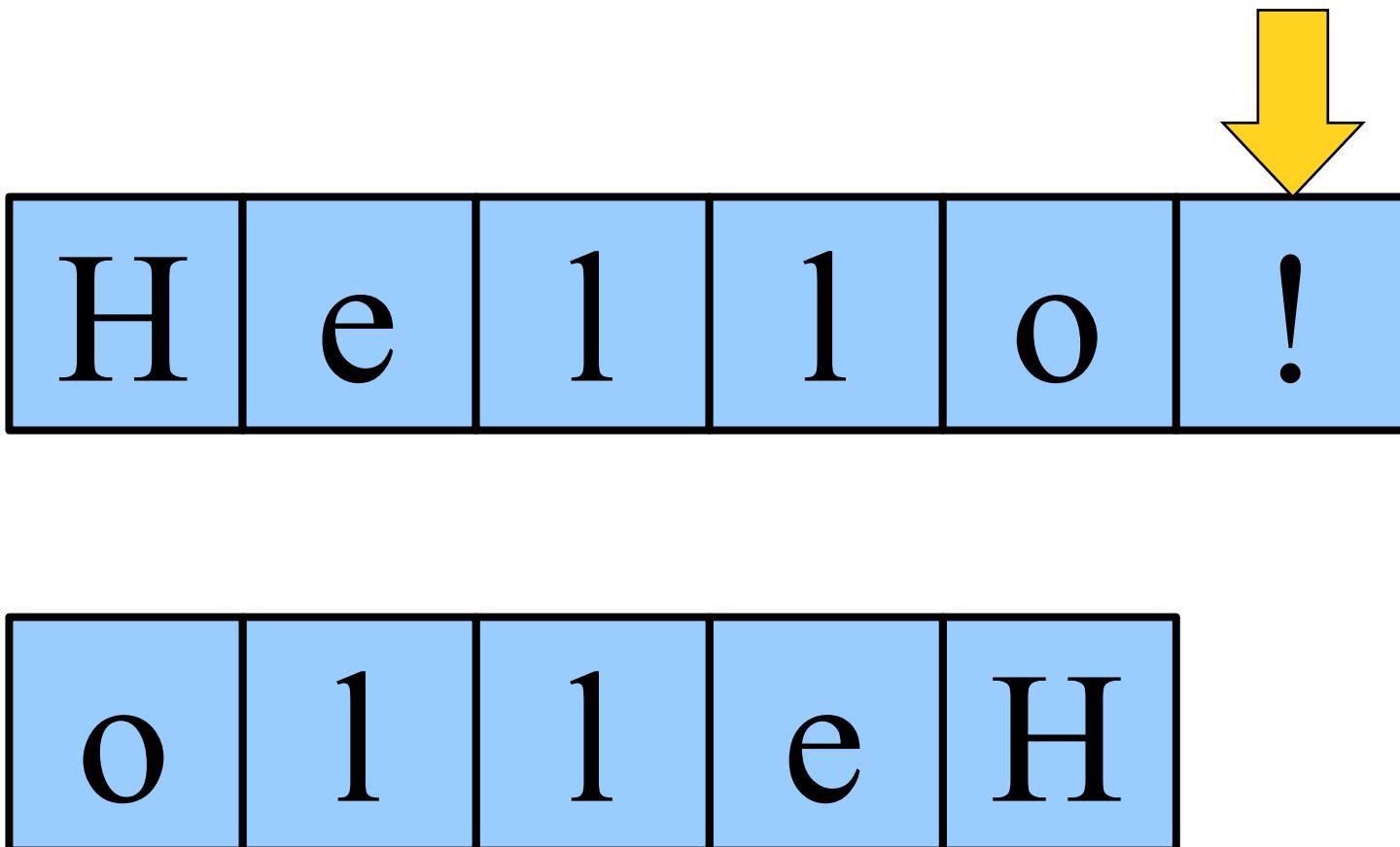
Reversing a String



Reversing a String



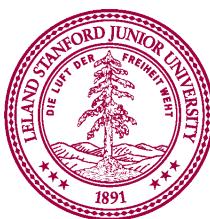
Reversing a String



Reversing a String

| | | | | | |
|---|---|---|---|---|---|
| H | e | l | l | o | ! |
|---|---|---|---|---|---|

| | | | | | |
|---|---|---|---|---|---|
| ! | o | l | l | e | H |
|---|---|---|---|---|---|



End Review

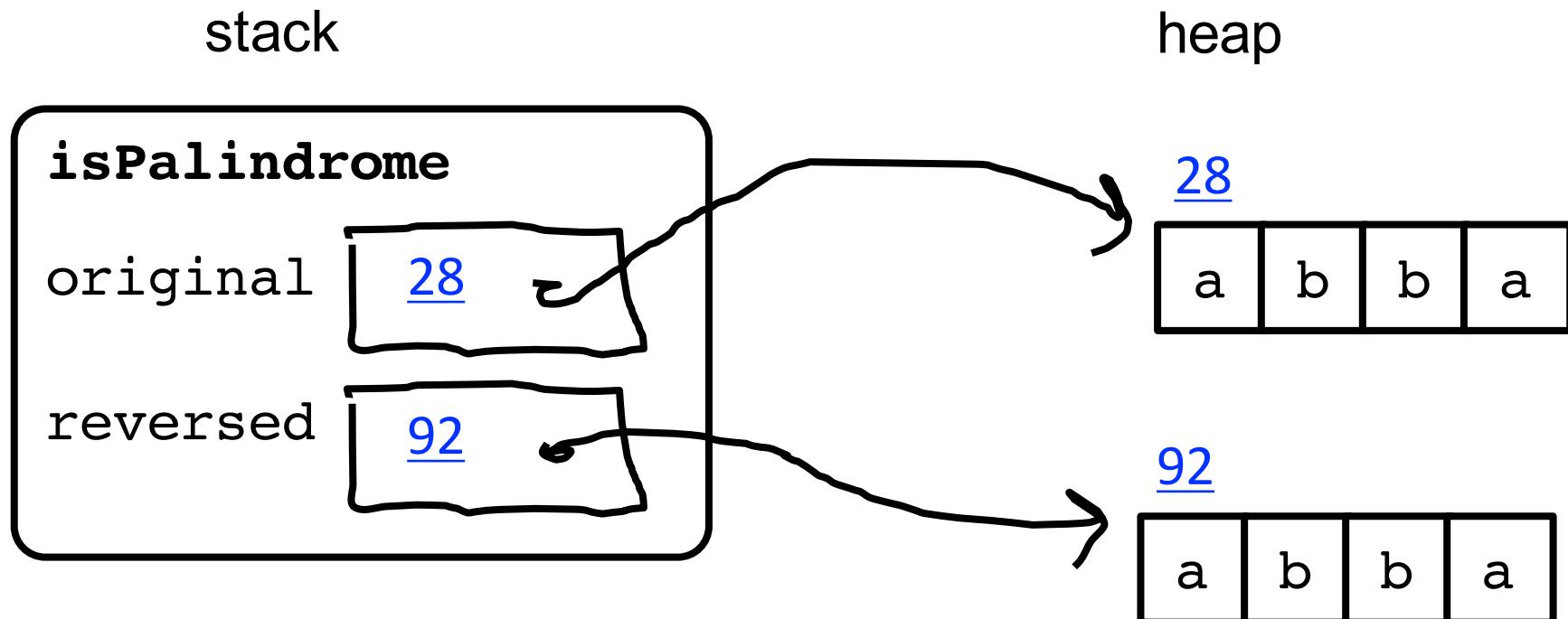
Palindrome

- A **palindrome** is a string that reads the same forwards and backwards.
- For example:
 - Abba
 - Racecar
 - Kayak
 - Mr. Owl ate my metal worm.
 - Go hang a salami! I'm a lasagna hog.
 - Elu par cette crapule



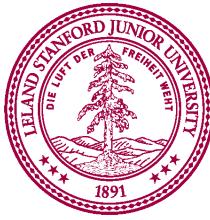
What went wrong?

```
private boolean isPalindrome(String original) {  
    String reversed = reverse(original);  
    return reversed == original;  
}
```





Use `.equals` to compare
strings, not `==`



Useful String methods

int length()

Returns the length of the string

char charAt(int index)

Returns the character at the specified index. Note: Strings indexed starting at 0.

String substring(int p1, int p2)

Returns the substring beginning at **p1** and extending up to but not including **p2**

String substring(int p1)

Returns substring beginning at **p1** and extending through end of string.

boolean equals(String s2)

Returns true if string **s2** is equal to the receiver string. This is case sensitive.

int compareTo(String s2)

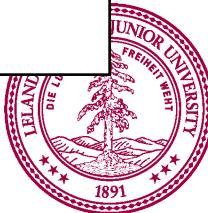
Returns integer whose sign indicates how strings compare in lexicographic order

int indexOf(char ch) or int indexOf(String s)

Returns index of first occurrence of the character or the string, or -1 if not found

String toLowerCase() or String toUpperCase()

Returns a lowercase or uppercase version of the receiver string



Some test cases

- Let's test our program on some examples:
 - Racecar
 - Kayak
 - Mr. Owl ate my metal worm.
 - Go hang a salami! I'm a lasagna hog.
- Will it work?



Stress Test

A man, a plan, a caret, a ban, a myriad, a sum, a lac, a liar, a hoop, a pint, a catalpa, a gas, an oil, a bird, a yell, a vat, a caw, a pax, a wag, a tax, a nay, a ram, a cap, a yam, a gay, a tsar, a wall, a car, a luger, a ward, a bin, a woman, a vassal, a wolf, a tuna, a nit, a pall, a fret, a watt, a bay, a daub, a tan, a cab, a datum, a gall, a hat, a tag, a zap, a say, a jaw, a lay, a wet, a gallop, a tug, a trot, a trap, a tram, a torr, a caper, a top, a tonk, a toll, a ball, a fair, a sax, a minim, a tenor, a bass, a passer, a capital, a rut, an amen, a ted, a cabal, a tang, a sun, an ass, a maw, a sag, a jam, a dam, a sub, a salt, an axon, a sail, an ad, a wadi, a radian, a room, a rood, a rip, a tad, a pariah, a revel, a reel, a reed, a pool, a plug, a pin, a peek, a parabola, a dog, a pat, a cud, a nu, a fan, a pal, a rum, a nod, an eta, a lag, an eel, a batik, a mug, a mot, a nap, a maxim, a mood, a leek, a grub, a gob, a gel, a drab, a citadel, a total, a cedar, a tap, a gag, a rat, a manor, a bar, a gal, a cola, a pap, a yaw, a tab, a raj, a gab, a nag, a pagan, a bag, a jar, a bat, a way, a papa, a local, a gar, a baron, a mat, a rag, a gap, a tar, a decal, a tot, a led, a tic, a bard, a leg, a bog, a burg, a keel, a doom, a mix, a map, an atom, a gum, a kit, a baleen, a gala, a ten, a don, a mural, a pan, a faun, a ducat, a pagoda, a lob, a rap, a keep, a nip, a gulp, a loop, a deer, a leer, a lever, a hair, a pad, a tapir, a door, a moor, an aid, a raid, a wad, an alias, an ox, an atlas, a bus, a madam, a jag, a saw, a mass, an anus, a gnat, a lab, a cadet, an em, a natural, a tip, a caress, a pass, a baronet, a minimax, a sari, a fall, a ballot, a knot, a pot, a rep, a carrot, a mart, a part, a tort, a gut, a poll, a gateway, a law, a jay, a sap, a zag, a tat, a hall, a gamut, a dab, a can, a tabu, a day, a batt, a waterfall, a patina, a nut, a flow, a lass, a van, a mow, a nib, a draw, a regular, a call, a war, a stay, a gam, a yap, a cam, a ray, an ax, a tag, a wax, a paw, a cat, a valley, a drib, a lion, a saga, a plat, a catnip, a pooh, a rail, a calamus, a dairyman, a bater, a canal – Panama!



Really test your programs!

Here are some palindromes in other languages:

- حلب بلاد قلعة تحدق في قلعة Halab (Dates hang underneath a castle in Halab)
- 여보, 안경 안보여 (Honey, I can't see my glasses)
- কড়ক (a loud thunderous sound)
- 上海自來水來自海上 (Shanghai tap water originates from "above" the ocean)

The comedian Dmitri Martin also has a routine about palindromes; check it out at
<https://www.youtube.com/watch?v=0hUHDIOazIU>



ATGCTTAAACC..

Human Genome Project

"ATGCCAGGAC"

"GGACTTACATTTTT"

"ATTTTTGGCCGGCC"

And how we remember that the human genome has
3 billion base pairs



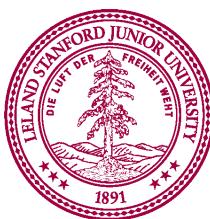
Compose Problem

strand1 "ATGCCAGGAC"

strand2 "GGACTTACATTTT"

result "ATGCCAGGACTTACATTTT"

And how we remember that the human genome has
3 billion base pairs



Ha. Gene was working on
The Genome project ☺



Did Gene Myers define all those
little pieces as constants?

File Processing!

Thanks Keith Schwarz for some great slides to build off!

Getting Data into Programs

- Put it directly in the program:
 - Define constants holding your values.
- Get it from the user:
 - Mouse events, readLine, etc.
- Generate it randomly:
 - Use a RandomGenerator.
- Get it from an external source.
 - Store it in a file and read it later.



Reading Files

- Virtually all programs that you've used at some point read files from disk:
 - Word processing (documents)
 - Web browser (cookies)
 - Games (saved progress)
 - Eclipse (Java files)
 - Music player (songs)



The structure of files

- A file is just a series of **bits** (ones and zeros).
- Those bits can have structure:
 - Plain-text: Bits represent characters.
 - JPEG: Bits encode information about the structure of an image.
 - MP3: Bits encode frequency information about music.
 - etc.



The structure of files

A file is just a series of *bits* (ones and zeros).

Those bits can have structure:

- Plain-text: Bits represent characters.

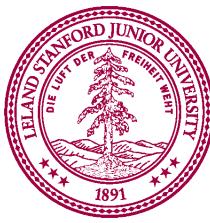
JPEG: Bits encode information about the structure of an image.

MP3: Bits encode frequency information about music.

etc.



Yesterday, upon the stair,
I met a man who wasn't there
He wasn't there again today
I wish, I wish he'd go away...
– Hughes Mearns, "Antagonish"



Yesterday, upon the stair,
I met a man who wasn't there
He wasn't there again today
I wish, I wish he'd go away...
– Hughes Mearns, "Antagonish"

Step one:
Open the file for reading.



Yesterday, upon the stair,
I met a man who wasn't there
He wasn't there again today
I wish, I wish he'd go away...
– Hughes Mearns, "Antagonish"

```
BufferedReader br =  
    new BufferedReader(new FileReader("poem.txt"));
```



Yesterday, upon the stair,
I met a man who wasn't there
He wasn't there again today
I wish, I wish he'd go away...
– Hughes Mearns, "Antagonish"

```
BufferedReader br =  
    new BufferedReader(new FileReader("poem.txt"));
```

To use the **BufferedReader** and **FileReader**
types, you need to

```
import java.io.*;
```



Yesterday, upon the stair,
I met a man who wasn't there
He wasn't there again today
I wish, I wish he'd go away...
- Hughes Mearns, "Antagonish"

```
BufferedReader br =  
    new BufferedReader(new FileReader("poem.txt"));
```



Yesterday, upon the stair,
I met a man who wasn't there
He wasn't there again today
I wish, I wish he'd go away...
- Hughes Mearns, "Antagonish"

```
BufferedReader br =  
    new BufferedReader(new FileReader("poem.txt"));
```



Yesterday, upon the stair,
I met a man who wasn't there
He wasn't there again today
I wish, I wish he'd go away...
- Hughes Mearns, "Antagonish"

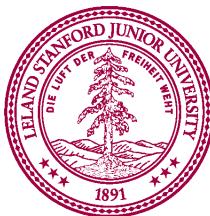
```
BufferedReader br =  
    new BufferedReader(new FileReader("poem.txt"));
```

Step Two:
Read the file,
one line at a time.



Yesterday, upon the stair,
I met a man who wasn't there
He wasn't there again today
I wish, I wish he'd go away...
- Hughes Mearns, "Antagonish"

```
BufferedReader br =  
    new BufferedReader(new FileReader("poem.txt"));  
  
String line1 = br.readLine();
```



Yesterday, upon the stair,

I met a man who wasn't there

He wasn't there again today

I wish, I wish he'd go away...

- Hughes Mearns, "Antagonish"

```
BufferedReader br =  
    new BufferedReader(new FileReader("poem.txt"));  
  
String line1 = br.readLine();
```



Yesterday, upon the stair,

I met a man who wasn't there

He wasn't there again today

I wish, I wish he'd go away...

- Hughes Mearns, "Antagonish"

```
BufferedReader br =  
    new BufferedReader(new FileReader("poem.txt"));
```

```
String line1 = br.readLine(); // Yesterday, upon the stair,
```



Yesterday, upon the stair,
I met a man who wasn't there
He wasn't there again today
I wish, I wish he'd go away...
- Hughes Mearns, "Antagonish"

```
BufferedReader br =  
    new BufferedReader(new FileReader("poem.txt"));  
  
String line1 = br.readLine(); // Yesterday, upon the stair,
```



Yesterday, upon the stair,
I met a man who wasn't there
He wasn't there again today
I wish, I wish he'd go away...
- Hughes Mearns, "Antagonish"

```
BufferedReader br =  
    new BufferedReader(new FileReader("poem.txt"));  
  
String line1 = br.readLine(); // Yesterday, upon the stair,  
String line2 = br.readLine();
```



Yesterday, upon the stair,
I met a man who wasn't there
He wasn't there again today
I wish, I wish he'd go away...
- Hughes Mearns, "Antagonish"

```
BufferedReader br =  
    new BufferedReader(new FileReader("poem.txt"));  
  
String line1 = br.readLine(); // Yesterday, upon the stair,  
String line2 = br.readLine();
```



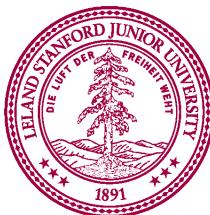
Yesterday, upon the stair,
I met a man who wasn't there
He wasn't there again today
I wish, I wish he'd go away...
- Hughes Mearns, "Antagonish"

```
BufferedReader br =  
    new BufferedReader(new FileReader("poem.txt"));  
  
String line1 = br.readLine(); // Yesterday, upon the stair,  
String line2 = br.readLine(); // I met a man who wasn't there
```



Yesterday, upon the stair,
I met a man who wasn't there
He wasn't there again today
I wish, I wish he'd go away...
- Hughes Mearns, "Antagonish"

```
BufferedReader br =  
    new BufferedReader(new FileReader("poem.txt"));  
  
String line1 = br.readLine(); // Yesterday, upon the stair,  
String line2 = br.readLine(); // I met a man who wasn't there
```



Yesterday, upon the stair,
I met a man who wasn't there
He wasn't there again today
I wish, I wish he'd go away...
- Hughes Mearns, "Antagonish"

```
BufferedReader br =  
    new BufferedReader(new FileReader("poem.txt"));  
  
String line1 = br.readLine(); // Yesterday, upon the stair,  
String line2 = br.readLine(); // I met a man who wasn't there  
String line3 = br.readLine();
```



Yesterday, upon the stair,
I met a man who wasn't there
He wasn't there again today
I wish, I wish he'd go away...
- Hughes Mearns, "Antagonish"

```
BufferedReader br =  
    new BufferedReader(new FileReader("poem.txt"));  
  
String line1 = br.readLine(); // Yesterday, upon the stair,  
String line2 = br.readLine(); // I met a man who wasn't there  
String line3 = br.readLine();
```



Yesterday, upon the stair,
I met a man who wasn't there
He wasn't there again today
I wish, I wish he'd go away...
- Hughes Mearns, "Antagonish"

```
BufferedReader br =  
    new BufferedReader(new FileReader("poem.txt"));  
  
String line1 = br.readLine(); // Yesterday, upon the stair,  
String line2 = br.readLine(); // I met a man who wasn't there  
String line3 = br.readLine(); // He wasn't there again today
```



Yesterday, upon the stair,
I met a man who wasn't there
He wasn't there again today
I wish, I wish he'd go away...
- Hughes Mearns, "Antagonish"

```
BufferedReader br =  
    new BufferedReader(new FileReader("poem.txt"));  
  
String line1 = br.readLine(); // Yesterday, upon the stair,  
String line2 = br.readLine(); // I met a man who wasn't there  
String line3 = br.readLine(); // He wasn't there again today
```



Yesterday, upon the stair,
I met a man who wasn't there
He wasn't there again today
I wish, I wish he'd go away...
- Hughes Mearns, "Antagonish"

```
BufferedReader br =  
    new BufferedReader(new FileReader("poem.txt"));  
  
String line1 = br.readLine(); // Yesterday, upon the stair,  
String line2 = br.readLine(); // I met a man who wasn't there  
String line3 = br.readLine(); // He wasn't there again today  
String line4 = br.readLine();
```



Yesterday, upon the stair,
I met a man who wasn't there
He wasn't there again today
I wish, I wish he'd go away...
- Hughes Mearns, "Antagonish"

```
BufferedReader br =  
    new BufferedReader(new FileReader("poem.txt"));  
  
String line1 = br.readLine(); // Yesterday, upon the stair,  
String line2 = br.readLine(); // I met a man who wasn't there  
String line3 = br.readLine(); // He wasn't there again today  
String line4 = br.readLine();
```



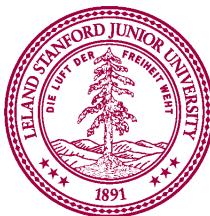
Yesterday, upon the stair,
I met a man who wasn't there
He wasn't there again today
I wish, I wish he'd go away...
- Hughes Mearns, "Antagonish"

```
BufferedReader br =  
    new BufferedReader(new FileReader("poem.txt"));  
  
String line1 = br.readLine(); // Yesterday, upon the stair,  
String line2 = br.readLine(); // I met a man who wasn't there  
String line3 = br.readLine(); // He wasn't there again today  
String line4 = br.readLine(); // I wish, I wish he'd go away...
```



Yesterday, upon the stair,
I met a man who wasn't there
He wasn't there again today
I wish, I wish he'd go away...
- Hughes Mearns, "Antagonish"

```
BufferedReader br =  
    new BufferedReader(new FileReader("poem.txt"));  
  
String line1 = br.readLine(); // Yesterday, upon the stair,  
String line2 = br.readLine(); // I met a man who wasn't there  
String line3 = br.readLine(); // He wasn't there again today  
String line4 = br.readLine(); // I wish, I wish he'd go away...
```



Yesterday, upon the stair,
I met a man who wasn't there
He wasn't there again today
I wish, I wish he'd go away...
- Hughes Mearns, "Antagonish"

```
BufferedReader br =  
    new BufferedReader(new FileReader("poem.txt"));  
  
String line1 = br.readLine(); // Yesterday, upon the stair,  
String line2 = br.readLine(); // I met a man who wasn't there  
String line3 = br.readLine(); // He wasn't there again today  
String line4 = br.readLine(); // I wish, I wish he'd go away...  
String line5 = br.readLine();
```



Yesterday, upon the stair,
I met a man who wasn't there
He wasn't there again today
I wish, I wish he'd go away...

— Hughes Mearns, "Antagonish"

```
BufferedReader br =  
    new BufferedReader(new FileReader("poem.txt"));  
  
String line1 = br.readLine(); // Yesterday, upon the stair,  
String line2 = br.readLine(); // I met a man who wasn't there  
String line3 = br.readLine(); // He wasn't there again today  
String line4 = br.readLine(); // I wish, I wish he'd go away...  
String line5 = br.readLine();
```



Yesterday, upon the stair,
I met a man who wasn't there
He wasn't there again today
I wish, I wish he'd go away...

- Hughes Mearns, "Antagonish"

```
BufferedReader br =  
    new BufferedReader(new FileReader("poem.txt"));  
  
String line1 = br.readLine(); // Yesterday, upon the stair,  
String line2 = br.readLine(); // I met a man who wasn't there  
String line3 = br.readLine(); // He wasn't there again today  
String line4 = br.readLine(); // I wish, I wish he'd go away...  
String line5 = br.readLine(); // - Hughes Mearns, "Antagonish"
```



Yesterday, upon the stair,
I met a man who wasn't there
He wasn't there again today
I wish, I wish he'd go away...
- Hughes Mearns, "Antagonish"

```
BufferedReader br =  
    new BufferedReader(new FileReader("poem.txt"));  
  
String line1 = br.readLine(); // Yesterday, upon the stair,  
String line2 = br.readLine(); // I met a man who wasn't there  
String line3 = br.readLine(); // He wasn't there again today  
String line4 = br.readLine(); // I wish, I wish he'd go away...  
String line5 = br.readLine(); // - Hughes Mearns, "Antagonish"
```



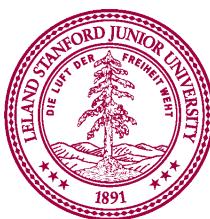
Yesterday, upon the stair,
I met a man who wasn't there
He wasn't there again today
I wish, I wish he'd go away...
- Hughes Mearns, "Antagonish"

```
BufferedReader br =  
    new BufferedReader(new FileReader("poem.txt"));  
  
String line1 = br.readLine(); // Yesterday, upon the stair,  
String line2 = br.readLine(); // I met a man who wasn't there  
String line3 = br.readLine(); // He wasn't there again today  
String line4 = br.readLine(); // I wish, I wish he'd go away...  
String line5 = br.readLine(); // - Hughes Mearns, "Antagonish"  
String line6 = br.readLine();
```



Yesterday, upon the stair,
I met a man who wasn't there
He wasn't there again today
I wish, I wish he'd go away...
- Hughes Mearns, "Antagonish"

```
BufferedReader br =  
    new BufferedReader(new FileReader("poem.txt"));  
  
String line1 = br.readLine(); // Yesterday, upon the stair,  
String line2 = br.readLine(); // I met a man who wasn't there  
String line3 = br.readLine(); // He wasn't there again today  
String line4 = br.readLine(); // I wish, I wish he'd go away...  
String line5 = br.readLine(); // - Hughes Mearns, "Antagonish"  
String line6 = br.readLine(); // *Returns null*
```



Yesterday, upon the stair,
I met a man who wasn't there
He wasn't there again today
I wish, I wish he'd go away...
- Hughes Mearns, "Antagonish"

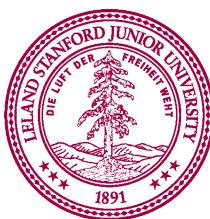
```
BufferedReader br =  
    new BufferedReader(new FileReader("poem.txt"));  
  
String line1 = br.readLine(); // Yesterday, upon the stair,  
String line2 = br.readLine(); // I met a man who wasn't there  
String line3 = br.readLine(); // He wasn't there again today  
String line4 = br.readLine(); // I wish, I wish he'd go away...  
String line5 = br.readLine(); // - Hughes Mearns, "Antagonish"  
String line6 = br.readLine(); // *Returns null*
```

Step Three:
Close the file.



Yesterday, upon the stair,
I met a man who wasn't there
He wasn't there again today
I wish, I wish he'd go away...
- Hughes Mearns, "Antagonish"

```
BufferedReader br =  
    new BufferedReader(new FileReader("poem.txt"));  
  
String line1 = br.readLine(); // Yesterday, upon the stair,  
String line2 = br.readLine(); // I met a man who wasn't there  
String line3 = br.readLine(); // He wasn't there again today  
String line4 = br.readLine(); // I wish, I wish he'd go away...  
String line5 = br.readLine(); // - Hughes Mearns, "Antagonish"  
String line6 = br.readLine(); // *Returns null*  
  
br.close();
```



Let “try” it out!

Thanks Keith Schwarz for some great slides to build off!

There's a "catch"

Thanks Keith Schwarz for some great slides to build off!

Sometimes things break

- Programs sometimes encounter unexpected errors.
- Sometimes these are bugs:
 - Dividing by zero.
 - Sending a message to a **null** object.
- Sometimes these are due to external factors:
 - Network errors.
 - Missing files.



Exceptional cases

- If Java encounters a case where it can't proceed as normal, it will cause an ***exception***.
- Java requires that your program handle certain types of exceptions.
- Think of exceptions as rerouting control in an emergency:
 - If all goes well, program continues as usual.
 - If something goes wrong, handle the emergency.



Let “try” it out!

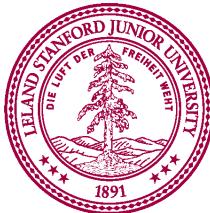
Thanks Keith Schwarz for some great slides to build off!

Let **try** it out!

Thanks Keith Schwarz for some great slides to build off!

try-ing your best

- To use a method or class that might cause an exception, you need to tell Java to **try** its best, knowing that it might fail.



try-ing your best

- To use a method or class that might cause an exception, you need to tell Java to **try** its best, knowing that it might fail.

```
BufferedReader br =  
    new BufferedReader(new FileReader("poem.txt"));  
  
String line1 = br.readLine(); // Yesterday, upon the stair,  
String line2 = br.readLine(); // I met a man who wasn't there  
String line3 = br.readLine(); // He wasn't there again today  
String line4 = br.readLine(); // I wish, I wish he'd go away...  
String line5 = br.readLine(); // - Hughes Mearns, "Antagonish"  
String line6 = br.readLine(); // *Returns null*  
  
br.close();
```



try-ing your best

- To use a method or class that might cause an exception, you need to tell Java to **try** its best, knowing that it might fail.

```
try {
    BufferedReader br =
        new BufferedReader(new FileReader("poem.txt"));

    String line1 = br.readLine(); // Yesterday, upon the stair,
    String line2 = br.readLine(); // I met a man who wasn't there
    String line3 = br.readLine(); // He wasn't there again today
    String line4 = br.readLine(); // I wish, I wish he'd go away...
    String line5 = br.readLine(); // - Hughes Mearns, "Antagonish"
    String line6 = br.readLine(); // *Returns null*

    br.close();
}
```



There's a "catch"

Thanks Keith Schwarz for some great slides to build off!

There's a **catch**

Thanks Keith Schwarz for some great slides to build off!

try and catch me

- If an exception occurs, you may need to tell Java to **catch** that exception.



try and catch me

- If an exception occurs, you may need to tell Java to **catch** that exception.

```
try {  
    BufferedReader br =  
        new BufferedReader(new FileReader("poem.txt"));  
  
    String line1 = br.readLine(); // Yesterday, upon the stair,  
    String line2 = br.readLine(); // I met a man who wasn't there  
    String line3 = br.readLine(); // He wasn't there again today  
    String line4 = br.readLine(); // I wish, I wish he'd go away...  
    String line5 = br.readLine(); // - Hughes Mearns, "Antagonish"  
    String line6 = br.readLine(); // *Returns null*  
  
    br.close();  
}
```



try and catch me

- If an exception occurs, you may need to tell Java to **catch** that exception.

```
try {
    BufferedReader br =
        new BufferedReader(new FileReader("poem.txt"));

    String line1 = br.readLine(); // Yesterday, upon the stair,
    String line2 = br.readLine(); // I met a man who wasn't there
    String line3 = br.readLine(); // He wasn't there again today
    String line4 = br.readLine(); // I wish, I wish he'd go away...
    String line5 = br.readLine(); // - Hughes Mearns, "Antagonish"
    String line6 = br.readLine(); // *Returns null*

    br.close();
} catch (IOException e) {
    println("An error occurred: " + e);
}
```



try and catch me

- If an exception occurs, you may need to handle that exception.

If something fails up here...

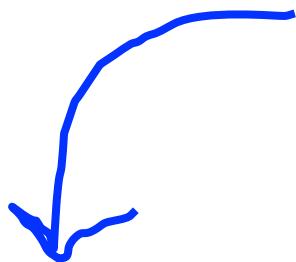
```
try {  
    BufferedReader br =  
        new BufferedReader(new FileReader("poem.txt"));  
  
    String line1 = br.readLine(); // Yesterday, upon the stair,  
    String line2 = br.readLine(); // I met a man who wasn't there  
    String line3 = br.readLine(); // He wasn't there again today  
    String line4 = br.readLine(); // I wish, I wish he'd go away...  
    String line5 = br.readLine(); // - Hughes Mearns, "Antagonish"  
    String line6 = br.readLine(); // *Returns null*  
  
    br.close();  
} catch (IOException e) {  
    println("An error occurred: " + e);  
}
```



try and catch me

```
try {  
    BufferedReader br =  
        new BufferedReader(new FileReader("poem.txt"));  
  
    String line1 = br.readLine(); // Yesterday, upon the stair,  
    String line2 = br.readLine(); // I met a man who wasn't there  
    String line3 = br.readLine(); // He wasn't there again today  
    String line4 = br.readLine(); // I wish, I wish he'd go away...  
    String line5 = br.readLine(); // - Hughes Mearns, "Antagonish"  
    String line6 = br.readLine(); // *Returns null*  
  
    br.close();  
} catch (IOException e) {  
    println("An error occurred: " + e);  
}
```

If something fails up here...



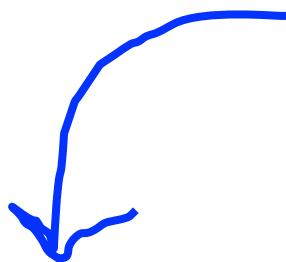
... we immediately jump down here.



try and catch me

```
try {  
    BufferedReader br =  
        new BufferedReader(new FileReader("poem.txt"));  
  
    String line1 = br.readLine(); // Yesterday, upon the stair,  
    String line2 = br.readLine(); // I met a man who wasn't there  
    String line3 = br.readLine(); // He wasn't there again today  
    String line4 = br.readLine(); // I wish, I wish he'd go away...  
    String line5 = br.readLine(); // - Hughes Mearns, "Antagonish"  
    String line6 = br.readLine(); // *Returns null*  
  
    br.close();  
} catch (IOException e) {  
    throw e;  
}
```

If something fails up here...



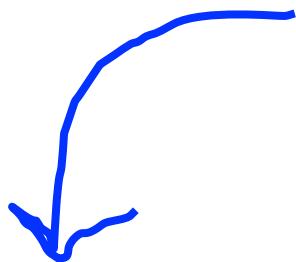
... we immediately jump down here.



try and catch me

```
try {  
    BufferedReader br =  
        new BufferedReader(new FileReader("poem.txt"));  
  
    String line1 = br.readLine(); // Yesterday, upon the stair,  
    String line2 = br.readLine(); // I met a man who wasn't there  
    String line3 = br.readLine(); // He wasn't there again today  
    String line4 = br.readLine(); // I wish, I wish he'd go away...  
    String line5 = br.readLine(); // - Hughes Mearns, "Antagonish"  
    String line6 = br.readLine(); // *Returns null*  
  
    br.close();  
} catch (IOException e) {  
    throw new RuntimeException(e);  
}
```

If something fails up here...



... we immediately jump down here.



File concepts in one slide

1. Make a BufferedReader (lets call it br) to open a file for reading

```
BufferedReader br = new BufferedReader(new FileReader("poem.txt"));
```

2. Use br.readLine to get one line from the file

```
br.readLine(); // returns the next line, or null
```

3. Both the above operations are “dangerous” so we need to use a try/catch loop

```
try{
    // live dangerously
} catch (Exception e){
    // have health insurance
}
```

4. You can either handle the problem or throw a runtime exception

```
throw new RuntimeException("AHHHH!");
```



lets **throw** it all together.

Thanks Keith Schwarz for some great slides to build off!

The classic file reading program.

- The idiomatic “read all the lines of a file” code is shown here:

```
try {
    BufferedReader br = /*...open the file... */
    while (true) {
        String line = br.readLine();
        if (line == null) break;

        /* ... process current line ... */
    }
    br.close();
} catch (IOException e) {
    throw new RuntimeException(e);
}
```



Understanding this code is about 95% of what we want you to know for files in CS106A



US Census Data

