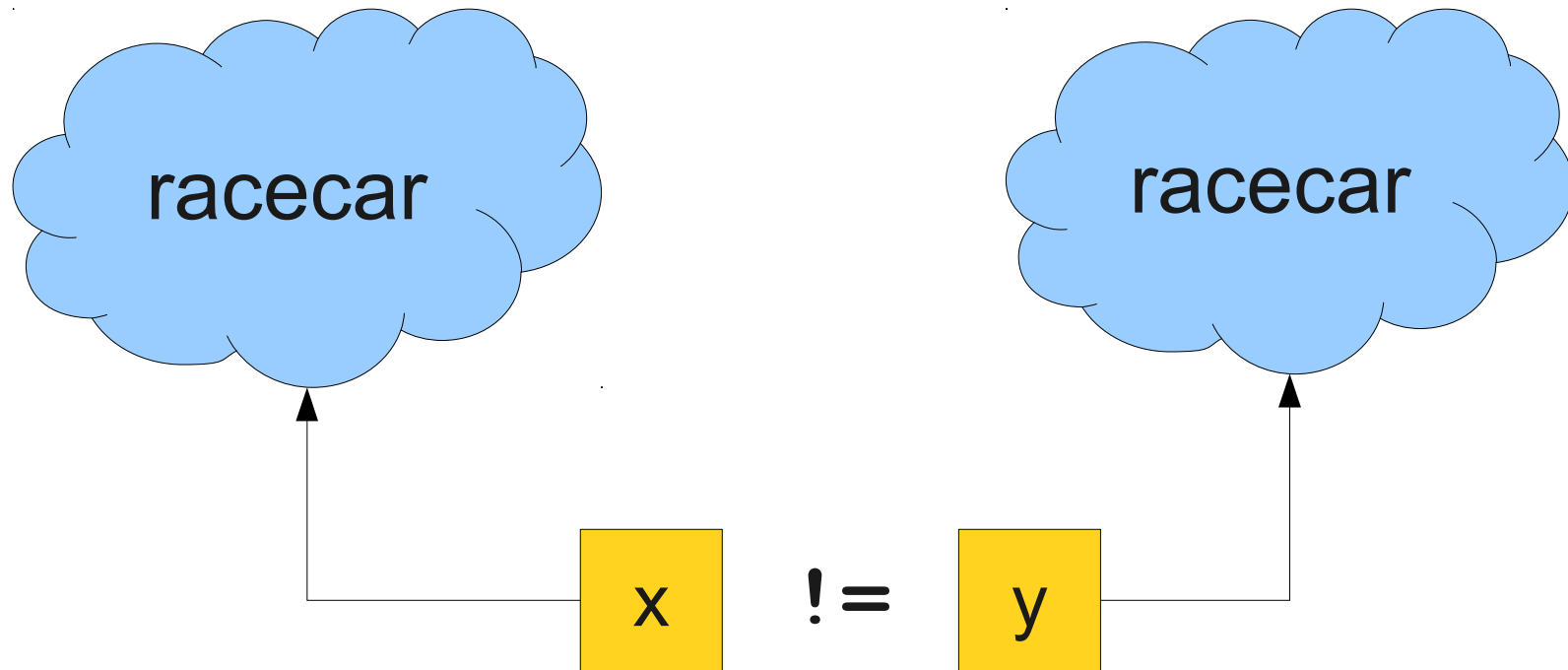


File Processing

Review from Last Time

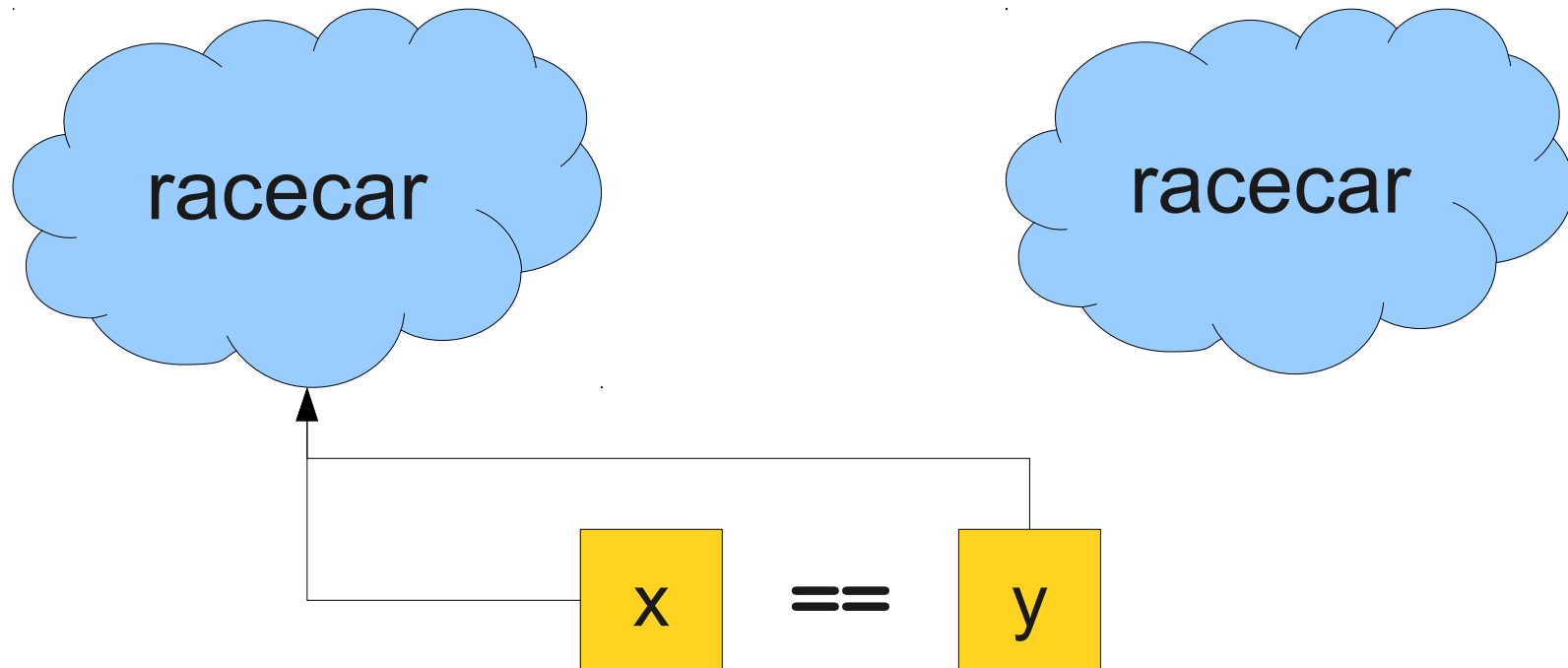
The == Operator

- When applied to objects, the == operator reports whether the two objects are the same object, not whether the *values* of those objects are equal.



The == Operator

- When applied to objects, the == operator reports whether the two objects are the same object, not whether the *values* of those objects are equal.



Comparing Strings for Equality

- To determine if two strings are equal, use the `.equals()` method:

```
String s1 = "racecar";  
String s2 = reverseString(s1);  
if (s1.equals(s2)) {  
    /* ... s1 and s2 are equal ... */  
}
```

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!

Searching a String

- You can search a string for a particular character or string by using the **indexOf** method:

***string*.indexOf(*pattern*)**

- **indexOf** returns the index of the first match if one exists.
- Otherwise, it returns -1 as a sentinel.

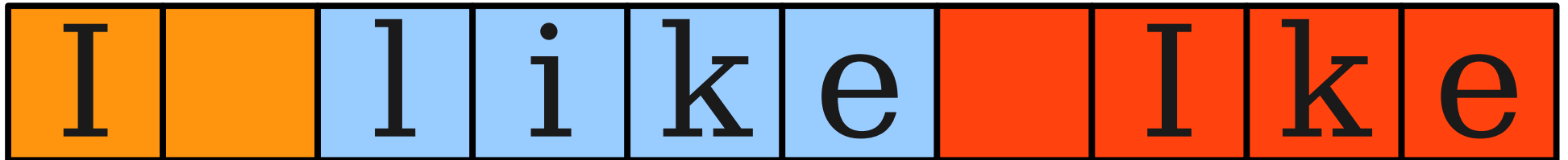
Replacing a Substring

- Because strings are immutable, it can be difficult to replace pieces of a string.
- To replace a segment of a string:
 - Obtain the **substring** of the string up to the point to replace.
 - Obtain the substring of the string after the point to replace.

I		l	i	k	e		I	k	e
---	--	---	---	---	---	--	---	---	---

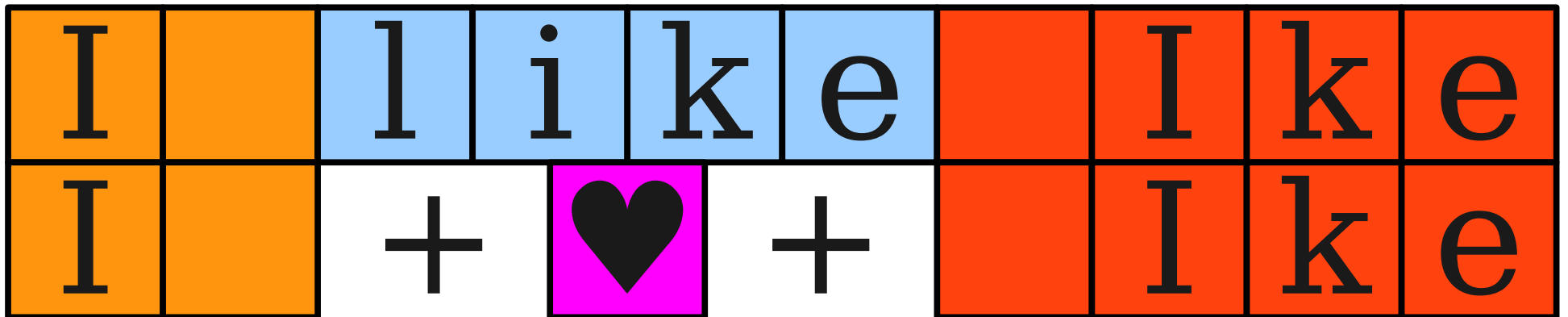
Replacing a Substring

- Because strings are immutable, it can be difficult to replace pieces of a string.
- To replace a segment of a string:
 - Obtain the **substring** of the string up to the point to replace.
 - Obtain the substring of the string after the point to replace.



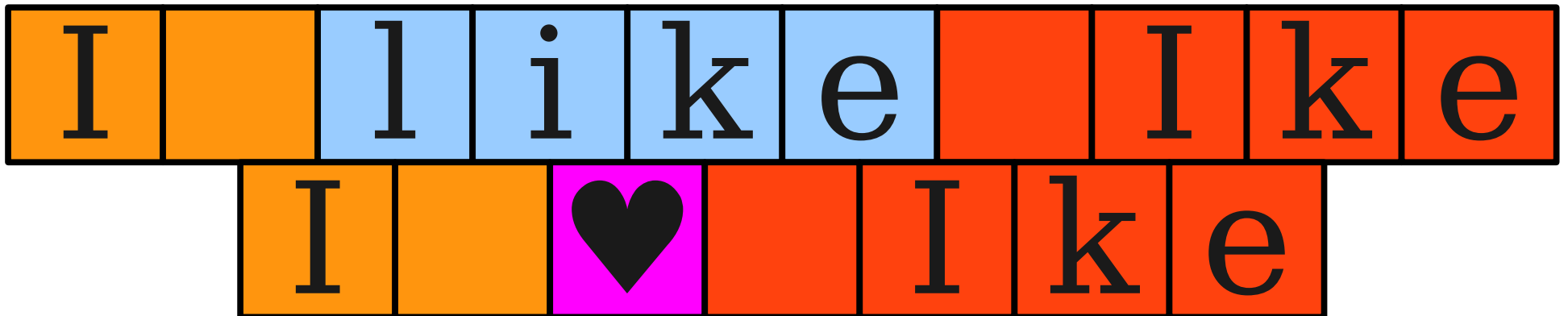
Replacing a Substring

- Because strings are immutable, it can be difficult to replace pieces of a string.
- To replace a segment of a string:
 - Obtain the **substring** of the string up to the point to replace.
 - Obtain the substring of the string after the point to replace.



Replacing a Substring

- Because strings are immutable, it can be difficult to replace pieces of a string.
- To replace a segment of a string:
 - Obtain the **substring** of the string up to the point to replace.
 - Obtain the substring of the string after the point to replace.





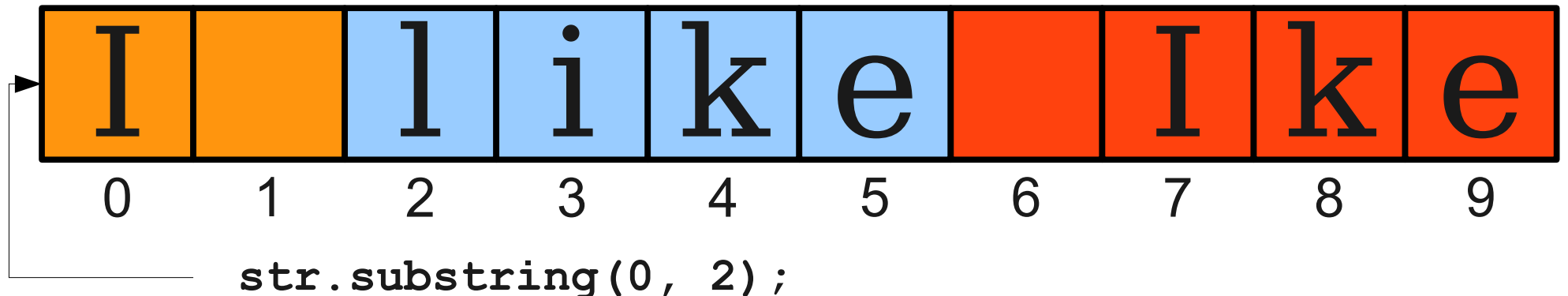
Obtaining Substrings

- To get all of the characters in the range [start, stop), use

string.substring(***start***, ***stop***)

- To get all of the characters from some specified point forward, use

string.substring(***start***)



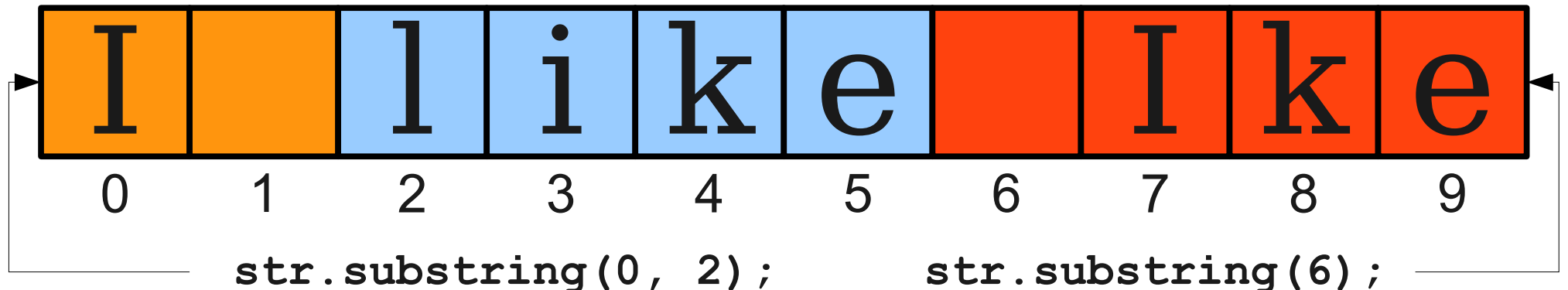
Obtaining Substrings

- To get all of the characters in the range [start, stop), use

***string*.substring(*start*, *stop*)**

- To get all of the characters from some specified point forward, use

***string*.substring(*start*)**



Time-Out for Announcements!

Midterm Date Change

- Midterm date changed to **Tuesday, February 11** from 7PM – 10PM.
 - *I am sorry for the inconvenience!*
 - Alternate exam times also changed; sign up ASAP if you can't make this time.
- To compensate, I've given everyone an extra late period this quarter.

Midterm Preparation

- Midterm review session this Sunday from 1PM - 4PM in Hewlett 200.
- Solutions to second practice exam released.
- Feel free to email the course staff with questions over the weekend!

Assignment 3

- Assignment 3 is due on Monday at 3:15PM.
- LaIR open Sunday night if you have questions.
- Feel free to email your section leader with questions!

Friday Four Square!

Today at 4:15PM, outside Gates

Back to CS106A!



Google

facebook®

Welcome to Big Data



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)
 - Eclipse (Java files)
 - Web browser (cookies)
 - IM client (stored login information)
 - Games (saved progress)
 - 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

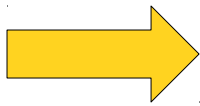
He wasn't there again today

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

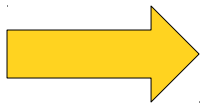
He wasn't there again today

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

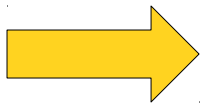
I wish, I wish he'd go away...

- 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,

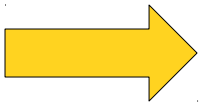
I met a man who wasn't there

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,

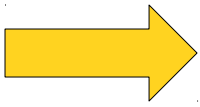
I met a man who wasn't there

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

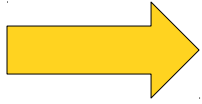
I wish, I wish he'd go away...

- Hughes Mearns, "Antagonish"

```
BufferedReader br =
```

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

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



Yesterday, upon the stair,

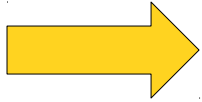
I met a man who wasn't there

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,

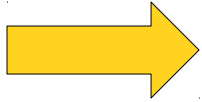
I met a man who wasn't there

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...
```

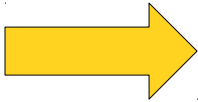
Yesterday, upon the stair,

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...
```

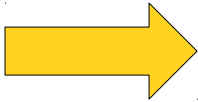

Yesterday, upon the stair,

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();
```

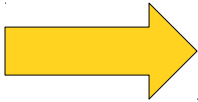
Yesterday, upon the stair,

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();
```

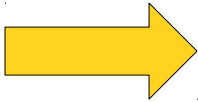
Yesterday, upon the stair,

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's Try It Out!

There's a Catch...

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's Try It Out!

Let's **try** It Out!

`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...

There's a **catch...**

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 tell Java to **catch** 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

- If an exception occurs, you may need to tell Java to **catch** 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);  
}
```

... we immediately jump down here.

Finally... let's make this program work!

Reading a File

- 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 line ... */
    }
    br.close();
} catch (IOException e) {
    /* ... handle error ... */
}
```

Fun with Data