

CSE 121 Lesson 3: Characters & Strings, Variables, Debugging

Matt Wang & Brett Wortzman

Autumn 2024



[sli.do #cse121](https://sli.do/#cse121)

TAs:

Abby	Afifah	Ailsa	Alice	Aliyan	Arohan
Chloë	Christopher	Dalton	Derek	Elizabeth	Ethan
Hanna	Hannah	Heather	Hibbah	Janvi	Jasmine
Judy	Julia	Kelsey	Lucas	Luke	Mahima
Maitreyi	Maria	Merav	Minh	Neha	Ronald
Ruslana	Sahej	Sam	Samrutha	Sushma	Vivian
Yijia	Zachary				

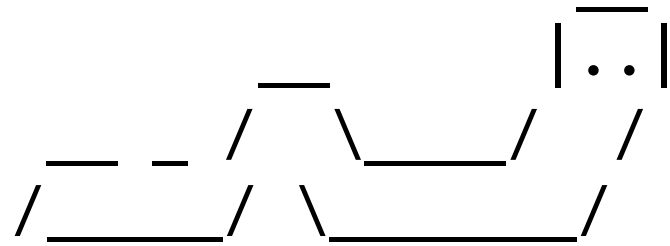
Today's playlist:
[121 24au lecture tunes](#)

Announcements, Reminders

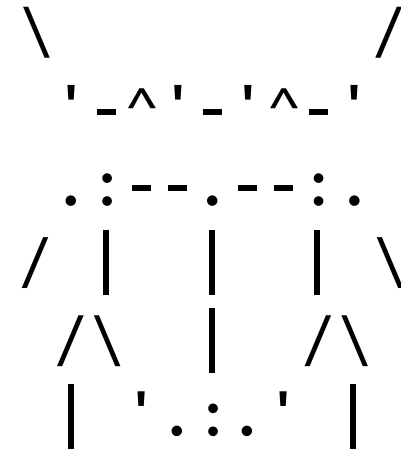
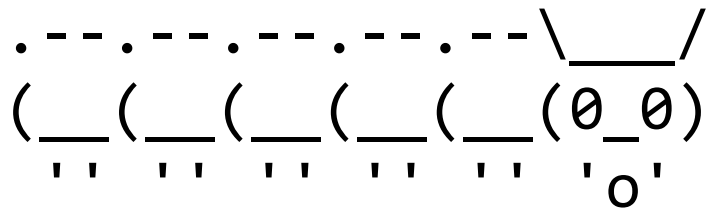
- P0 was released on Wed and is due Tues, Oct 8th
- Quiz 0 scheduled for Oct 24th (about 3 weeks away)
 - More details will be released in the coming weeks!
 - Prep includes practice quizzes, section materials, etc.
- Helpful website resources
 - [Grading Rubrics](#) (P0 rubric will be out tonight!)
 - [Ed shortcuts](#)
 - Search Site button!

Beautiful Bugs (1/x)

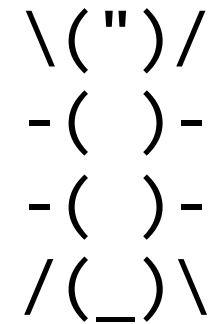
Wow! After all that rain,
a worm came up!



Caterpillar :0



A beetle just passed by...



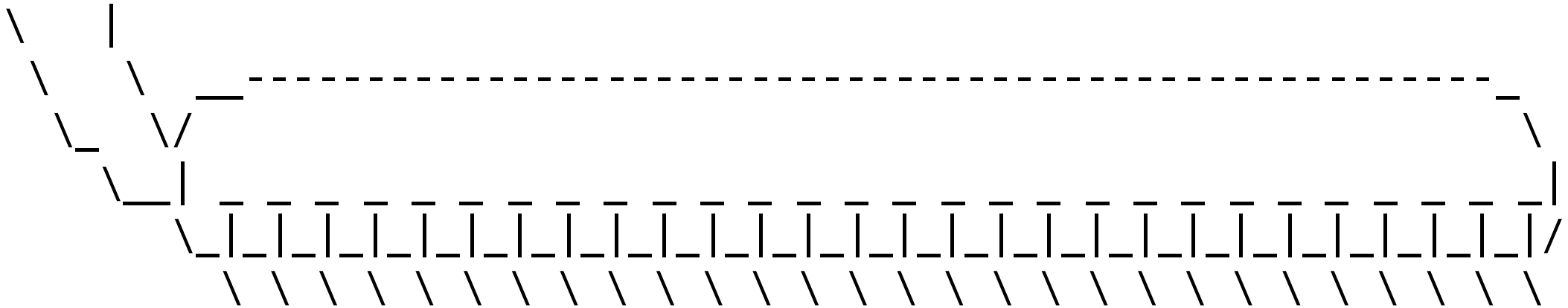
This is a cute ant.

Beautiful Bugs (3/x)

A Centipede EWW

```
(((((  
oooooo<  
(((
```

That centipede better not exceed 100 characters in length!



Milipede Mervin:

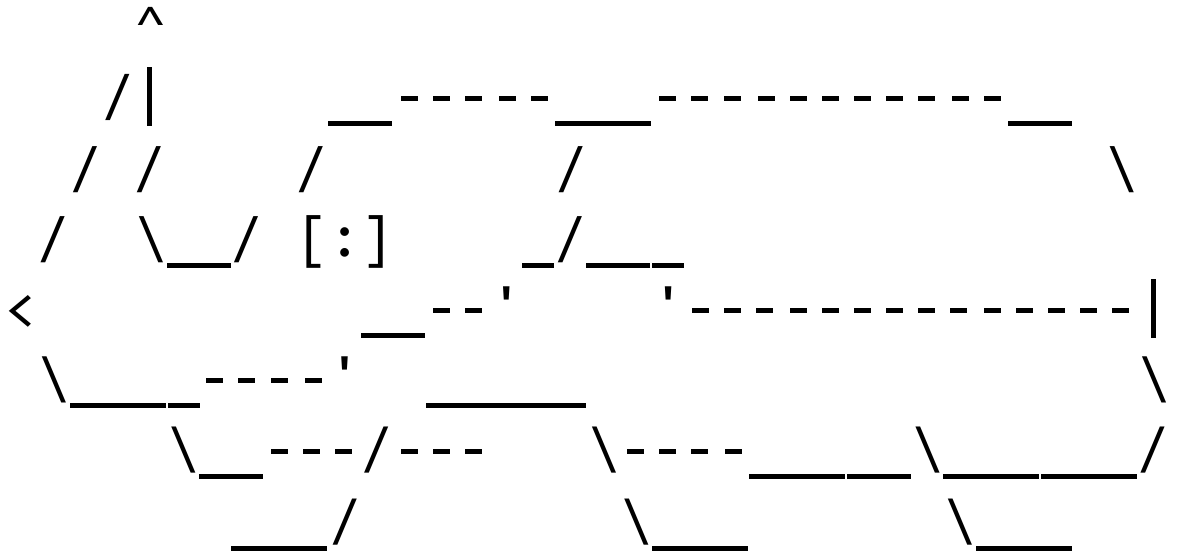


Centipede!

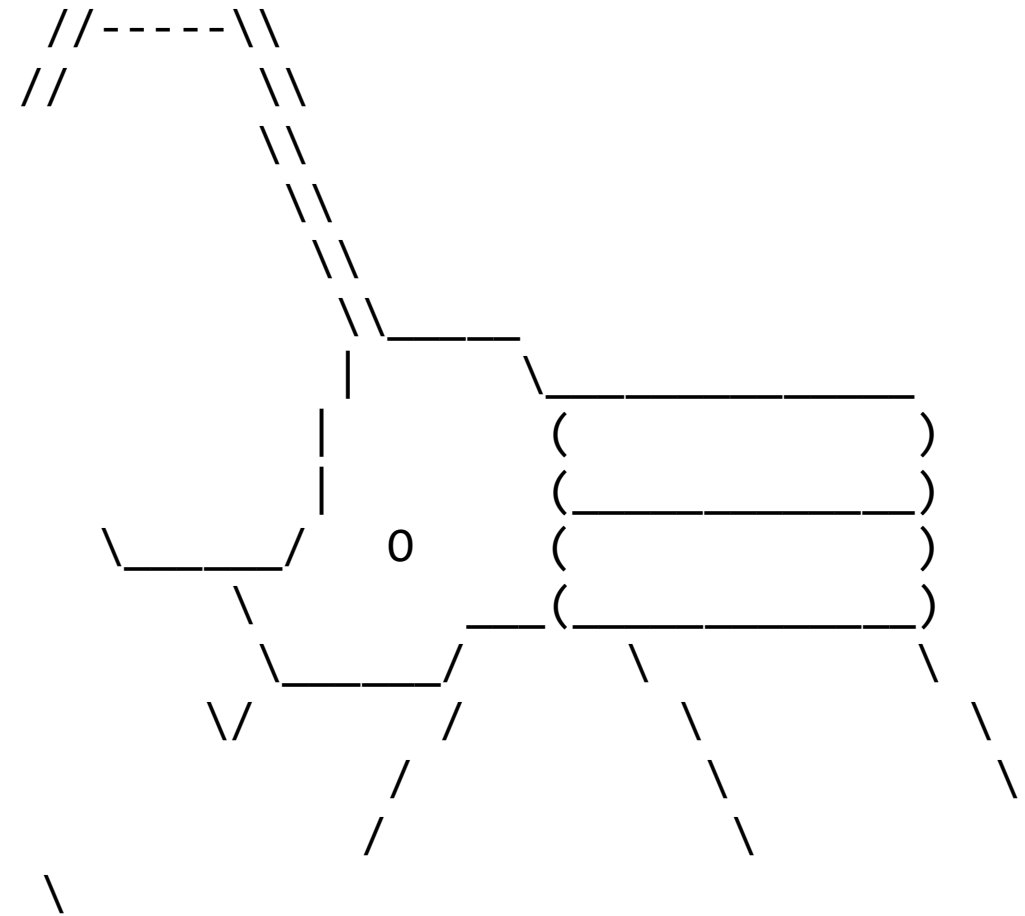


Beautiful Bugs (4/x)

Is that a rhino or a beetle?
It's a rhino beetle!

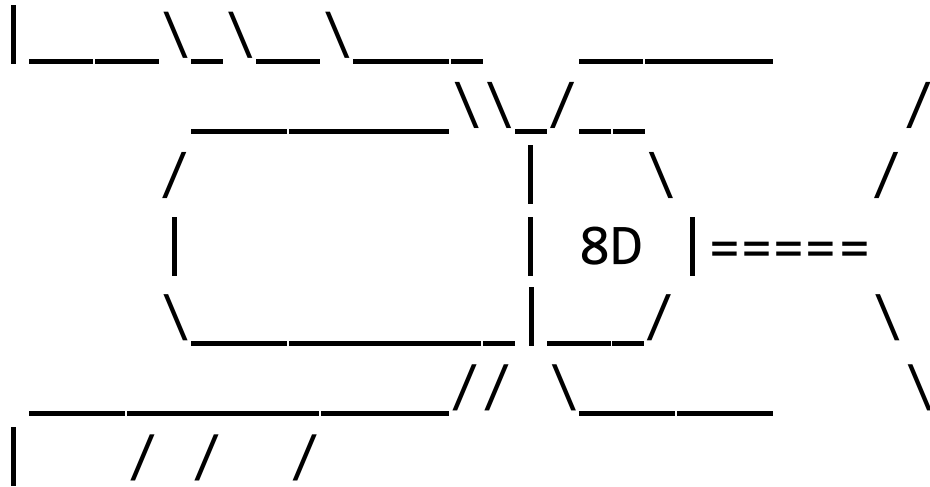


RHINO BEETLE ☹(ò_ó~)☹

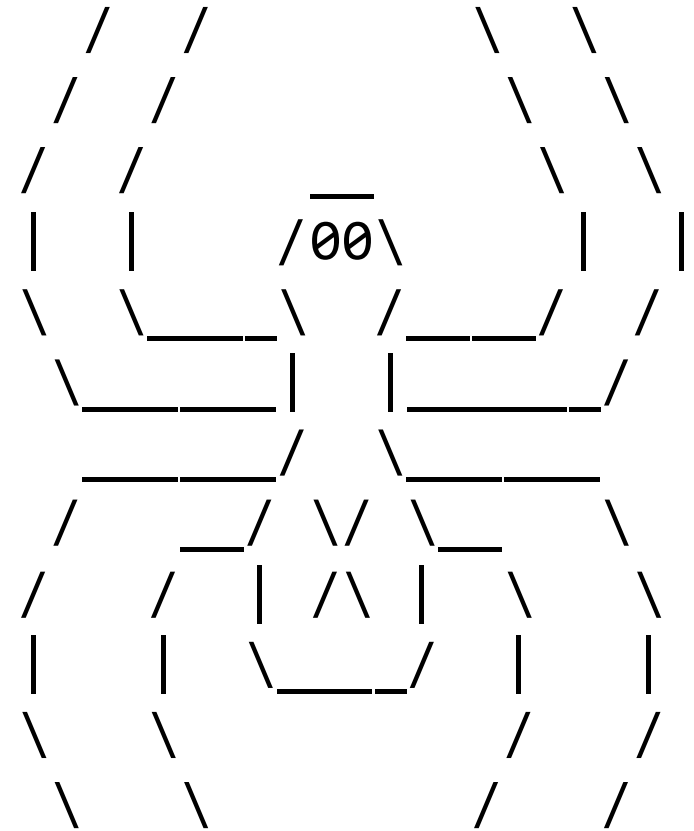


Beautiful Bugs (5/x)

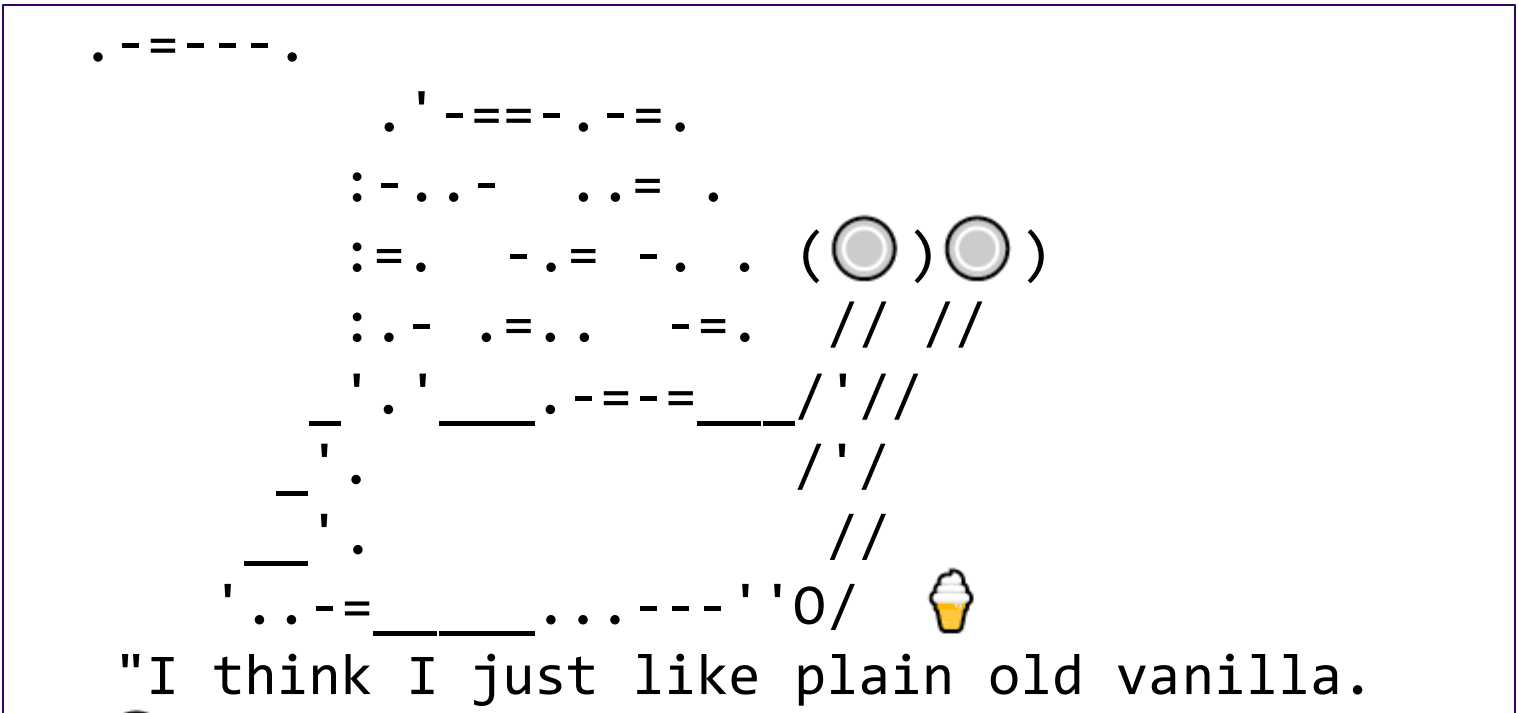
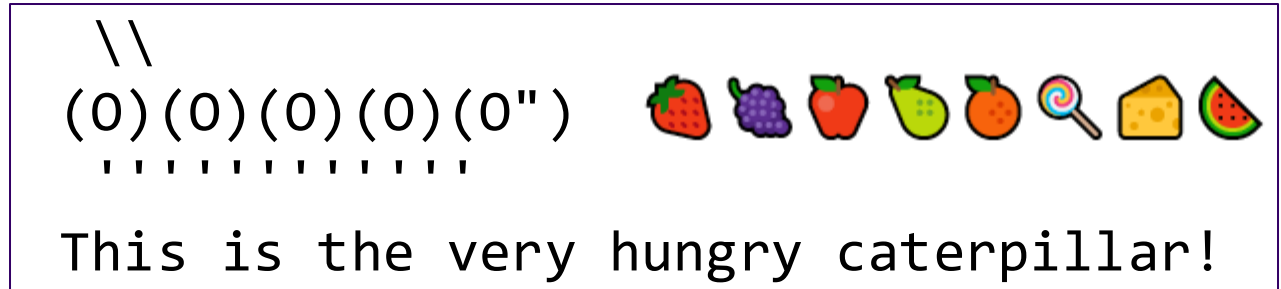
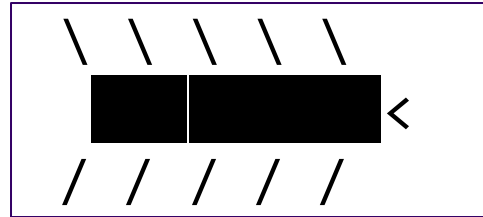
OH NO! Is that the evil roach??



No problem! Spider(man) is here!!

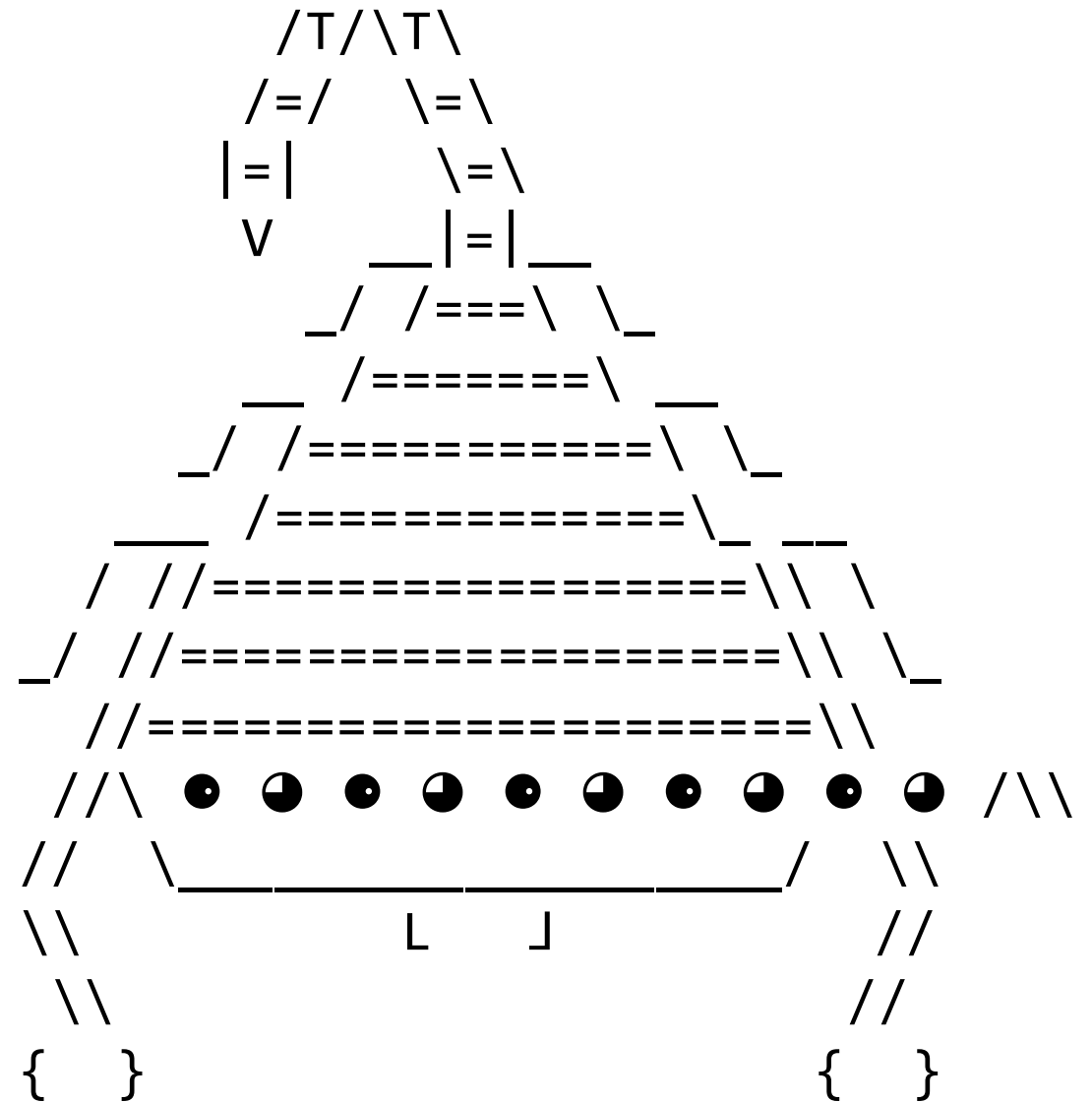
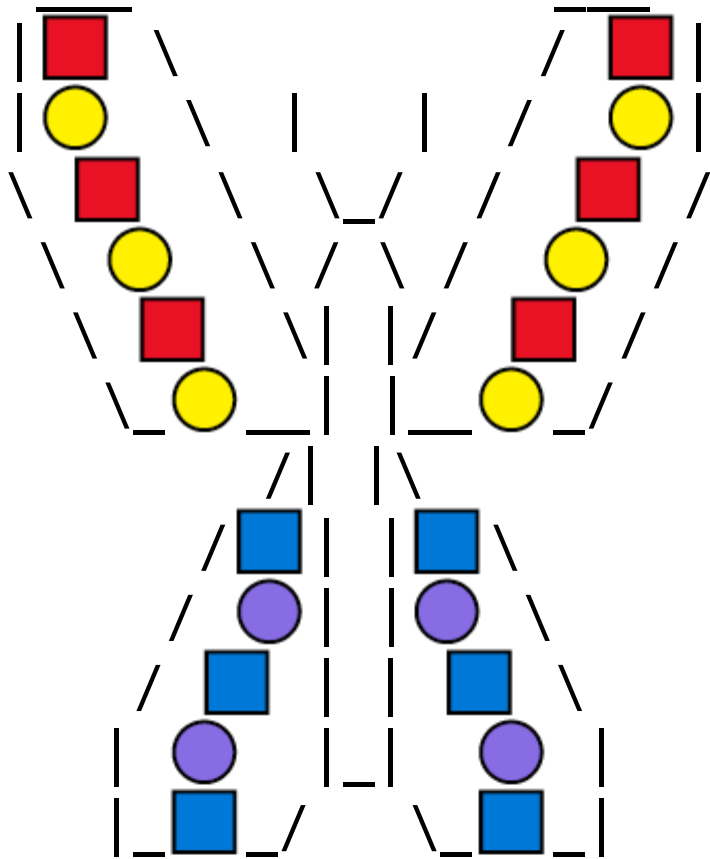


Beautiful Bugs (6/x)



Beautiful Bugs (7/x)

A vibrant butterfly!



SUPER SCORPION is here to save the day!

Beautiful Bugs (8/x)

Hey! Look! A scarab with his dung ball

```

屎屎屎
屎屎屎屎屎屎屎--虫虫虫虫
屎屎屎屎屎屎屎屎屎—虫虫虫虫
屎屎屎屎屎屎屎屎屎    虫虫虫    虫虫
屎屎屎屎屎屎屎屎屎    \    虫虫虫虫虫
屎屎屎屎屎屎屎屎屎    /    \ \    虫
屎屎屎                    /    \ \

```



Beautiful Bugs (9/x)

OH NO, and its the worst bug of them all!...

```
HelloBugs.java:31: error:HelloBugs.java:31: error:  
HelloBugs.java:31: error:HelloBugs.java:31: error:HelloBugs.java:31: error:  
HelloBugs.java:31: error:HelloBugs.java:31: error:HelloBugs.java:31  
HelloBugs.java:31: error:HelloBugs.java:31: error:HelloBugs.java:31  
HelloBugs.java:31: error:HelloBugs.java:31: error:HelloBugs.java:31: error:  
HelloBugs.java:31: error:HelloBugs.java:31: error:
```

And on accessibility... (1/2)

Loved your reflection responses here! Some common themes:

- not previously knowing how blind people code (or use computers)
 - screenreaders are *very* fast
 - keyboard-only navigation seems challenging
 - debugging (and programming) is already so hard!
- emphasizing that accessibility (and inclusion) matters!
 - “it’s the small things that matter”
 - companies should care too!
- very impressed by the speaker’s perseverance and determination
 - follow-up: is this fair?

And on accessibility... (2/2)

We then asked you: **is C0 accessible for a blind developer?**

Broad spectrum of answers, but **most of you said no.** Some reasons:

- in looking at ASCII art:
 - the caption is not enough context for a blind user
 - the screenreader would not interact well with the actual art
 - caption could be low-quality or wrong!
- *doing* the assignment would be even harder
 - how to implement the various requirements (especially spacing?)
 - does Ed have the same accessibility features? (Matt says: no)
 - assignment may be fundamentally inaccessible!

So, what?

Broadly speaking: the digital world is inaccessible (but that's changing)! Making things accessible requires **active** effort & learning.

In CSE 121, we don't have the full knowledge yet to make accessible ASCII art (or Java programs, applications, video games, websites, ...)

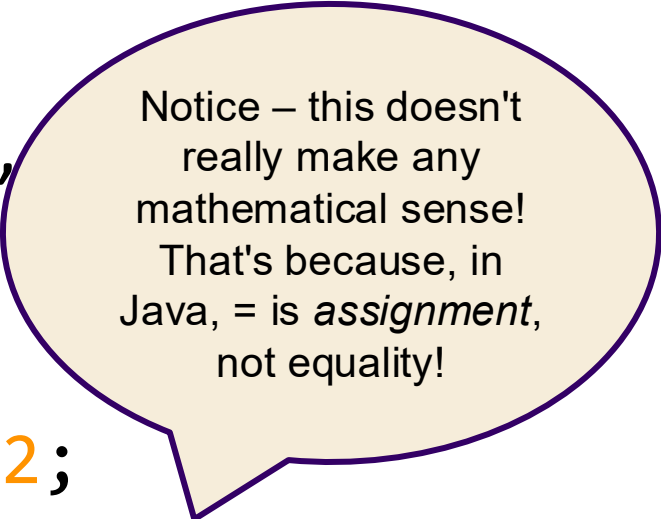
However, we encourage you to:

- think about accessibility when you make things with computers
- keep on learning more! UW is a **global leaders** in digital accessibility
 - e.g. at UW: [CSE 493E: Accessibility](#), [CREATE](#), [AccessComputing](#)

Variables – Manipulation

They're made to be manipulated, modified,

```
int myFavoriteNumber = 7;  
int doubleFV = myFavoriteNumber * 2;  
myFavoriteNumber = myFavoriteNumber + 3;
```



Notice – this doesn't really make any mathematical sense! That's because, in Java, = is *assignment*, not equality!

New Operators! (1/3)

```
myFavoriteNumber = myFavoriteNumber + 3;
```

This pattern is so common, we have a shorthand for it!

```
myFavoriteNumber += 3;
```

Note: this works for both numeric addition and string concatenation!

New Operators! (2/3)

The shorthands `-=`, `*=`, `/=`, and `%=` exist too!

```
myFavoriteNumber /= 3;
```

Should this work for integers? Doubles? Strings?

New Operators! (3/3)

There are even shorter operators for “incrementing” and “decrementing”!

```
myFavoriteNumber++; // equal to myFavoriteNumber += 1;  
myFavoriteNumber--; // equal to myFavoriteNumber -= 1;
```

Should this work for integers? Doubles? Strings?

Poll in with your answer!



[sli.do #cse121](https://sli.do/#cse121)

What do a, b, and c hold after this code is executed?

```
int a = 10;  
int b = 30;  
int c = a + b;  
c -= 10;  
a = b + 5;  
b /= 2;
```

A. 10, 30, 40

B. 35, 15, 30

C. 35, 15.5, 30

D. 20, 15, 30

Poll in with your answer!



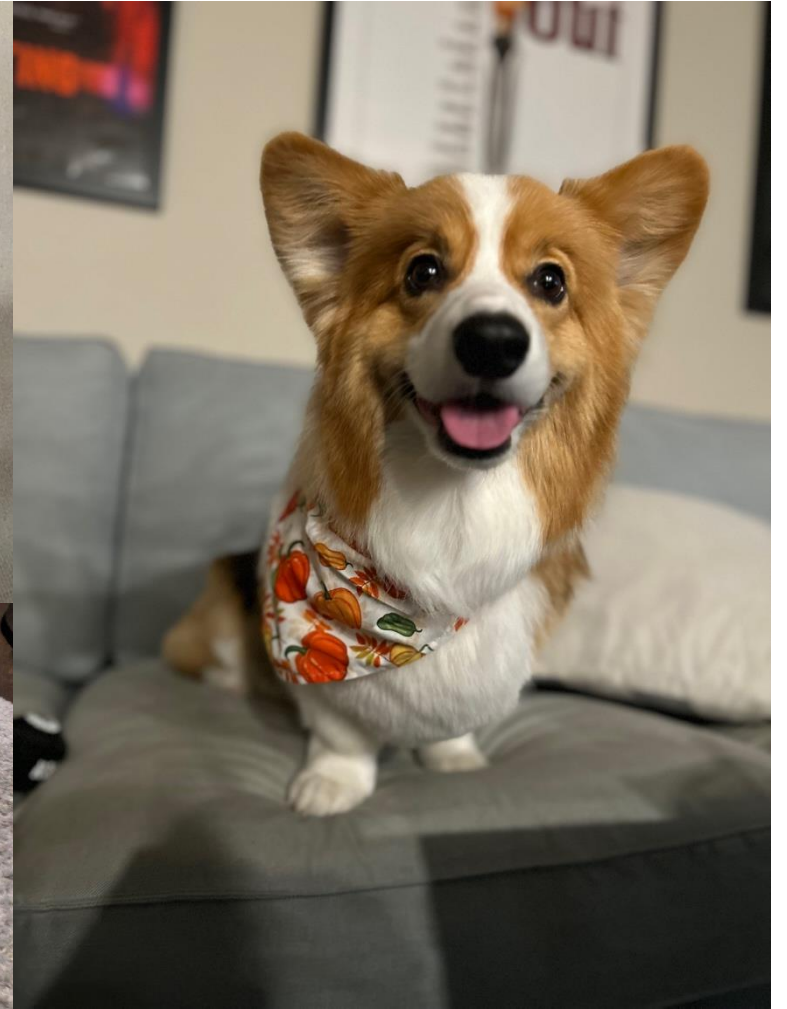
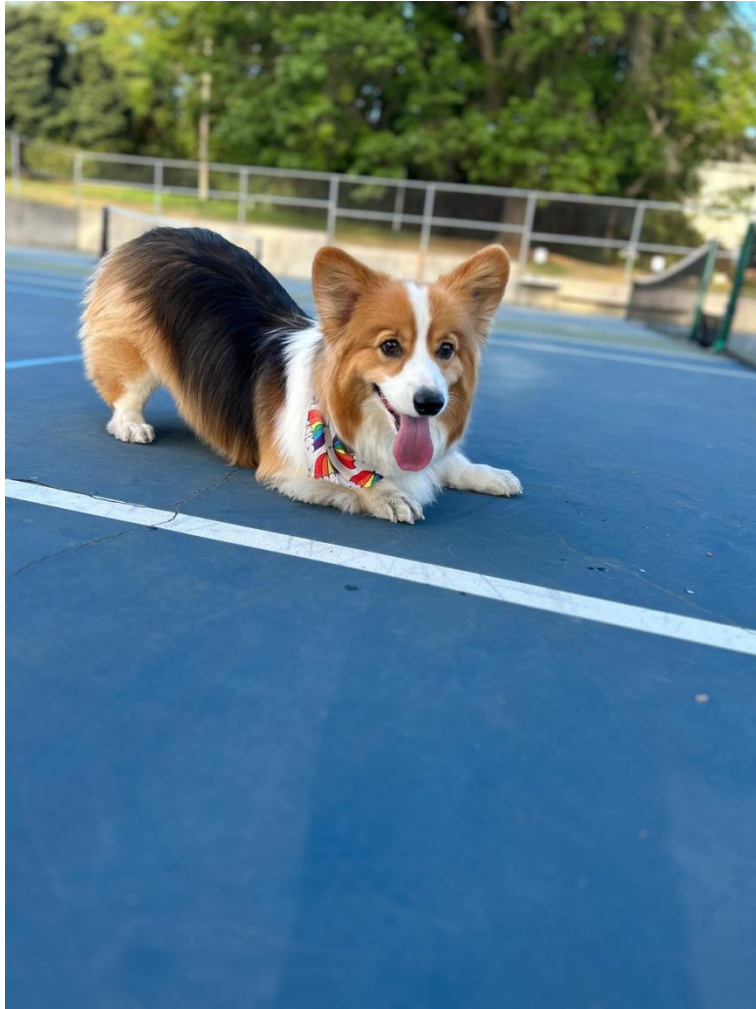
[sli.do #cse121](https://sli.do/#cse121)

Suppose `s` contains the String "bubble gum". Which option below would result in `s` containing "Gumball" instead?

b	u	b	b	l	e		g	u	m
0	1	2	3	4	5	6	7	8	9

- A. `s.substring(7) + "ball";`
- B. `s = s.substring(7, 9) + "ball";`
- C. `s = s.charAt(7).toUpperCase() + "ball";`
- D. `s = s.substring(7, 8).toUpperCase() + s.substring(8) + "ball";`

Interlude: Gumball



(PCM) String Methods

Usage: `<string variable>.<method>(…)`

Method	Description
<code>length()</code>	Returns the length of the string.
<code>charAt(i)</code>	Returns the character at index <i>i</i> of the string
<code>indexOf(s)</code>	Returns the index of the first occurrence of <i>s</i> in the string; returns -1 if <i>s</i> doesn't appear in the string
<code>substring(i, j)</code> or <code>substring(i)</code>	Returns the characters in this string from <i>i</i> (inclusive) to <i>j</i> (exclusive); if <i>j</i> is omitted, goes until the end of the string
<code>contains(s)</code>	Returns whether or not the string contains <i>s</i>
<code>equals(s)</code>	Returns whether or not the string is equal to <i>s</i> (case-sensitive)
<code>equalsIgnoreCase(s)</code>	Returns whether or not the string is equal to <i>s</i> ignoring case
<code>toUpperCase()</code>	Returns an uppercase version of the string
<code>toLowerCase()</code>	Returns a lowercase version of the string

Reminders

- P0 was released on Wed and is due Tues, Oct 8th
- Quiz 0 scheduled for Oct 24th (about 3 weeks away)
- Helpful website resources
 - [Grading Rubrics](#) (P0 rubric will be out tonight!)
 - [Ed shortcuts](#)
 - Search Site button!
- Questions? Come to office hours!
 - today: Matt (12:30-1:20) + IPL until 6:30