

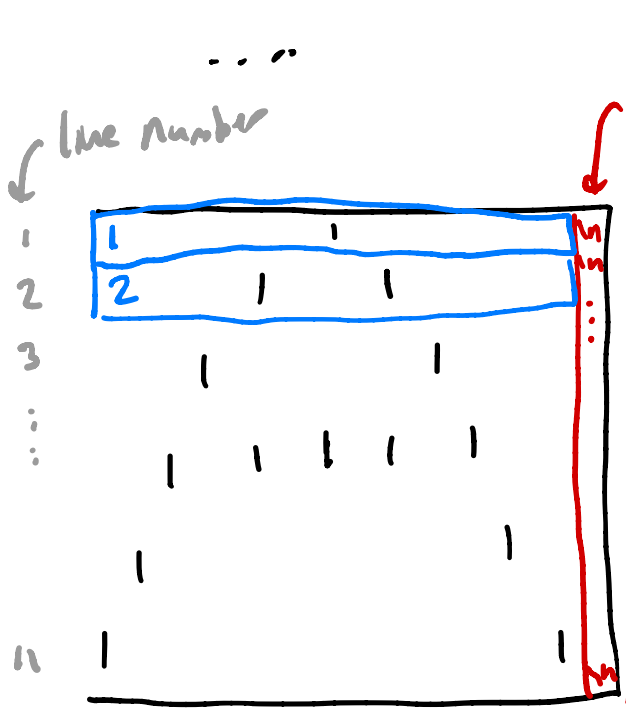
Lecture 24 | ASCII Art

font Alphabet.txt

A B C D ... a b c d ... ? !
64 total

alphabet :
all symbols
we know
how to
convert
from string
character
to c++.

font Symbols.txt



all symbols take up
11 lines

new lines are invisible

len = 15 characters long

len = 14 without newline

11 lines

B

Step 1 Fill in dictionary

Suggested way to organize Symbols:

dictionary d

64 Keys

Keys

Single chars in ASCII art font alphabet.

e.g. 'A', 'B', ..., 'a', 'b', ...

Values

List of 11 lines that make up each symbol

$d['A'] = [11 \text{ items} - 11 \text{ lines/strings making up A symbol}]$

line number

1										
2										
3										
...										
...										
...										
...										
...										
...										
...										
11										

ascii list =

line 1

line 2

line 11

Strings len = 14

len(asciiList) = 11

$d['A'] = [\text{line 1}, \text{line 2}, \dots, \text{line 11}]^{14}$

$d['B'] = [\text{line 1}, \text{line 2}, \dots, \text{line 11}]^{14}$

String?

$d['c'] = \dots$

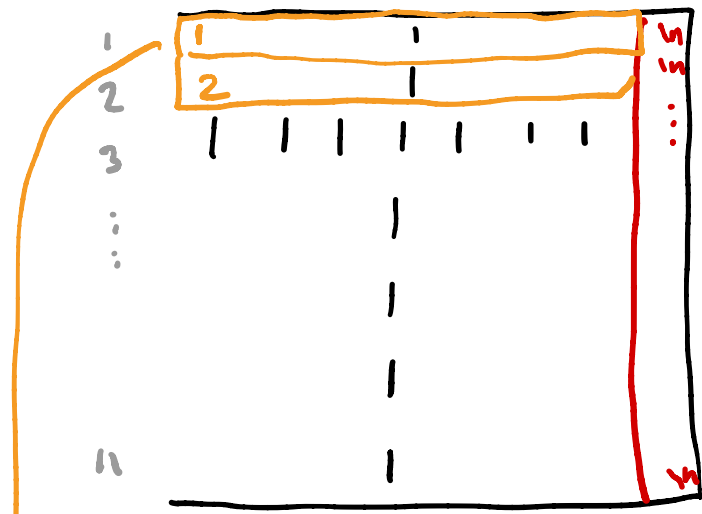
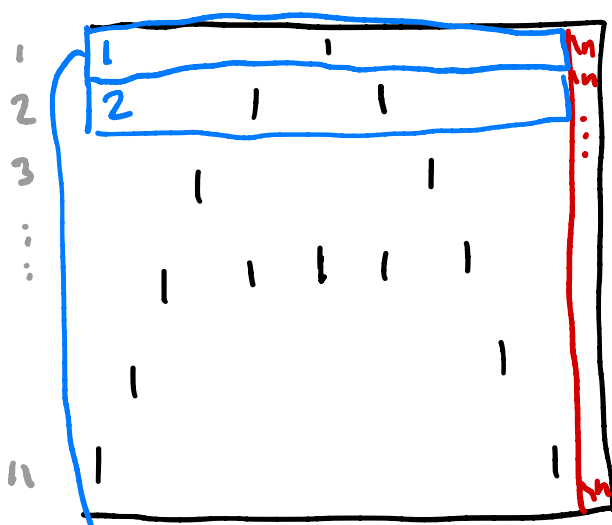
$\vdots$

$d['!'] = [\text{line 1}, \text{line 2}, \dots, \text{line 11}]^{14}$

$\text{len}(d.\text{keys}()) = 64 : \underbrace{A-Z, a-z, 0-9, !, ?}_{64 \text{ total}}$

Step 2 Building ASCII art String
by concatenating rows of each
appropriate symbol in the dictionary.

Example: Convert 'A+' to ASCII art.



string
 $\text{art} =$
1
 $+$
1
 $+ '\backslash n'$
 $+$
2
 $+$
2
 $+ '\backslash n'$
 $+ \dots$

A little more code detail :

$d['A'] = [11 \text{ lines making up } A]$

$d['t'] = [11 \text{ lines making up } t]$

$\text{art} = d['A'][0] + d['t'][0] + '\backslash n'$
 $+ d['A'][1] + d['t'][1] + '\backslash n'$
 $+ \dots$

+ d['A'] [10] + d['t'] [10]

Classes

So far: we have worked with Turtle and Zelle Objects.

★ Now: we will make our own kind of objects.

Goal Make Object type that represents a College Student — Student

Our own Constructor

like myCircle = gr.circle(gr.particle(0,0), 10)

jane = Student('jane') } calling the constructor

bob = Student('bob')

joe = Student('joe')

We have a Student class

returns a new Student object

Student Objects — have unique data
3 unique/different objects

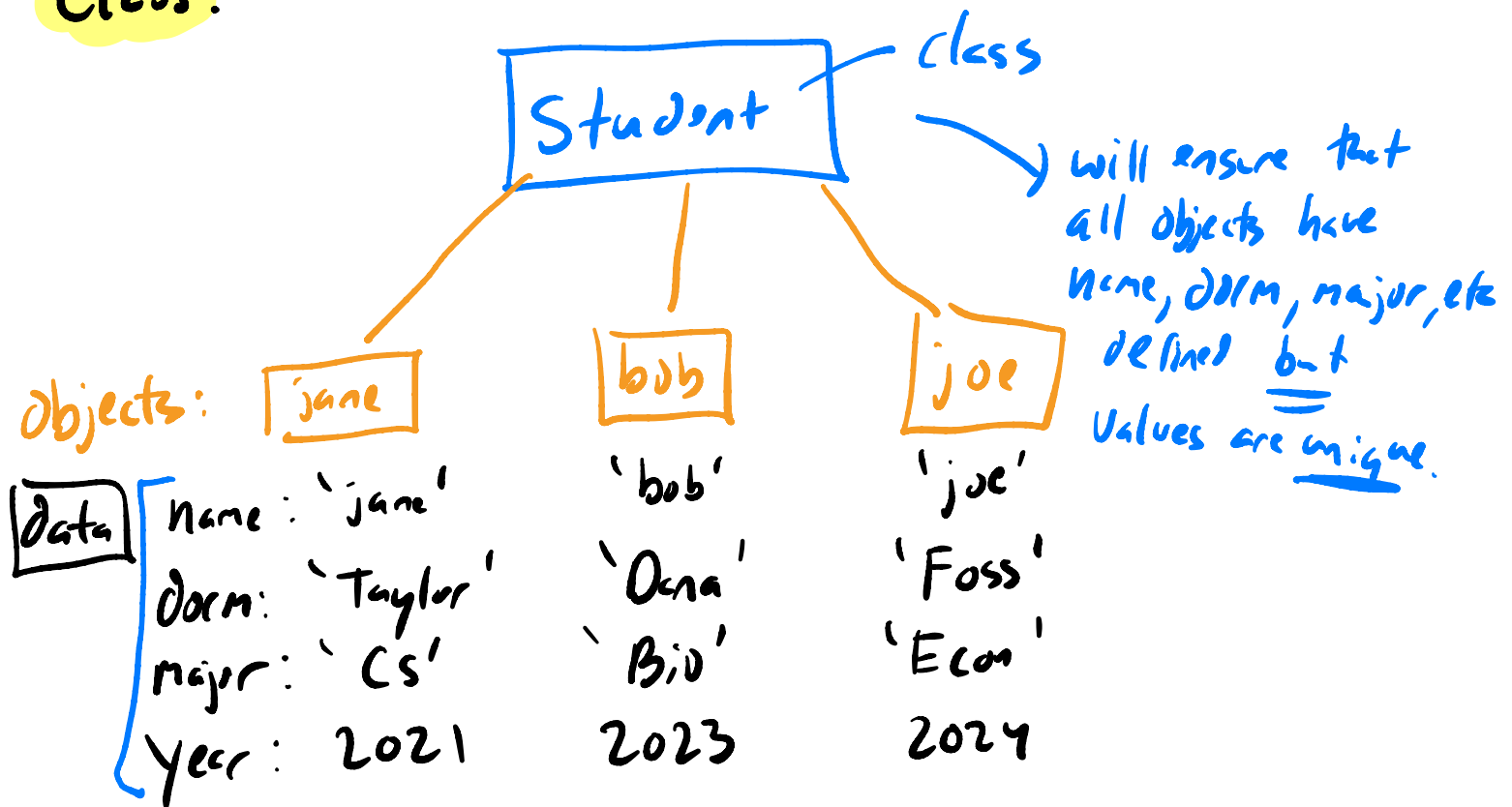
We often create object types from nouns
— "things"

e.g. Student, turtle, Screen, rectangle.

* Verbs/actions usually are methods

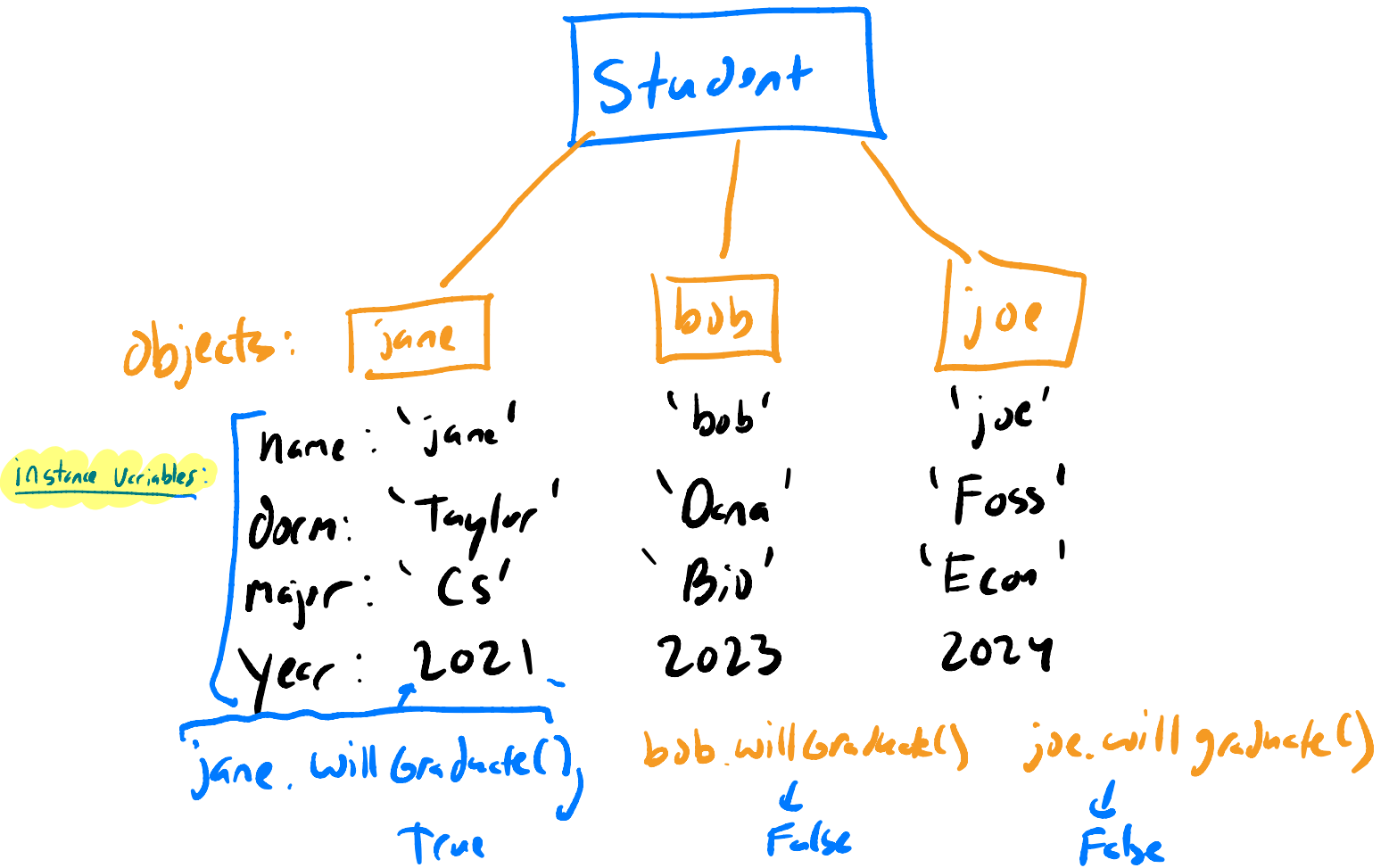
1) Classes allow us to store unique info for each object

An object type that we make is called a Class.



instance Variables: every object of Student class has these, but different values.

2) classes define actions we can take on any object using the instance variables:



We define our own methods

Common kinds of methods

1) Access data in an object — get method

e.g. jane.get Dorm(), jane.get Major(), jane.get Name()
return ('Taylor' 'CS' 'jane')

⇒ usually, get methods return something [instance var]

2) Change data in an object — Set methods

e.g. jane.setMajor('Math')


major: ~~'CS'~~ → 'Math'

⇒ Set methods usually don't return anything,
but have 1+ parameters.