Lecture 24: ASCII ART Overview

- **font Alphabet.txt**
  - A B C D ... a b c d ... ?!
  - 64 in total
  - alphabet: all symbols we can convert to ASCII art.

- **font Symbols.txt**
  - line number
  - other symbols

<table>
<thead>
<tr>
<th>1</th>
<th>width = 15 chars</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>

Last column = newline characters (invisible)

Suggested way to organize each ASCII art symbol:

**Dictionary**

**Keys:** Single characters in ASCII font
- e.g. 'A', 'B', 'C', ...

**Values:** List of 11 strings making up each of the 11 lines of each symbol.
• Read in each line of ASCII symbol as a string without new line characters.
• Append 11 consecutive line strings into a list:
  - Each list has 11 items (strings)
  - len(asciiList) = 11
• Add the list composing the 11 line strings for each ASCII symbol to a dictionary:
  - d[‘A’] = [14, 14, 14, …]
  - d[‘B’] = [14, 14, 14, …]
  - d[‘C’] = [14, 14, 14, …]
  - …
  - d[‘! ’] = [14, 14, 14, …]

len(d.keys()): 64  A-Z, a-z, 0-9, ?, !

# Keys = # font symbols in alphabet
Step 2: Concatenate rows of ASCII symbol strings representing characters of the input string one row at a time.

Example: Convert input string ‘A+’ to ASCII art.

```
String

art = 1 + 1 + \n + 2 + 2 + \n + ...
```
How to build these rows of the ASCII art?

\[ \text{art} = \text{d ['A'][0]} + \text{d ['t'][0]} + \text{\textbackslash n}
+ \text{d ['A'][1]} + \text{d ['t'][1]} + \text{\textbackslash n}
+ \text{...}
+ \text{d ['A'][10]} + \text{d ['t'][10]} + \text{\textbackslash n} \]
Lecture 25: Classes

A class allows us to create our own object types.

E.g. Not just use what turtle has to offer — opens up tons of possibilities!

Let's say we wanted to make a type of object for a Colby student — Student

Having Student class allows us to make objects of that type:

```python
Our class like calling
rect = gr. Circle(gr. Point(0,0), 10)

jane = Student(‘Jane’)  # name = ‘jane’
bob = Student(‘Bob’)  # name = ‘bob’
joe = Student(‘joe’)  # name = ‘joe’
```

Constructor: Special method that returns a new object — e.g. of type Student.
E.g. jane, bob, joe
3 new objects

Each object has unique data type Student.
What is a good thing to create objects of?

Nouns: things — e.g. Student is a thing
rectangle is a thing
turtle is a thing

Why make a class for a thing?

1) Groups data/information related to unique instances of that thing

Class

<table>
<thead>
<tr>
<th>Student</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jane</td>
</tr>
<tr>
<td>Bob</td>
</tr>
<tr>
<td>Joe</td>
</tr>
</tbody>
</table>

Objects

- Jane
  - name: Jane
  - age: 18
  - gpa: 4.0

- Bob
  - name: Bob
  - age: 20
  - gpa: 3.8

- Joe
  - name: Joe
  - age: 22
  - gpa: 3.6

Each object instance has same variables (name, age), but unique values. Called instance variables.

- An instance variable could be basic data type or an object
- It will be defined for any object you make of a class
- e.g. any Student object will have name, age, gpa variables.
More examples:

e.g. a Car class would probably have instance variables for: make and color

- all cars have these properties but of course specific values across cars could be different

- all cars have a make and color
- These 3 Cars may have different specific colors and makes

Example: 2D
circle class
circ1

\[
\begin{array}{c|c|c|c|c|c|c}
\text{circ1} & \text{circ2} \\
\hline
x & 0 & 11 \\
y & 0 & 12 \\
radius & 10 & 9 \\
\end{array}
\]

define what data any circle should have.
2) Classes allow us to perform actions on each instance, with its own data: methods

- we could add a method to our `Student` class to compute each student's GPA — `Compute GPA()`

```
Common types of methods:
- Access data from each instance of a class (object)
  - get method — method has "get" in it.
    - e.g. `get GPA()`, `get Name()`, `get Age()`
```

```
Change data from an instance — method has "set" in it.

E.g. set Age(), set GPA(), set Name()

It’s Bob’s birthday, we call Bob. set Age(21) to update age 20 → 21.

Let’s write a simple class for a student and play around with it.

Let’s analyze the memory model for a program involving Student class