Inheritance Revisited
Same Logic, Different Data Structure
Same Logic, Different Data Structure

- Search tree for the maze:
  - Check directions in order of:
    - North
    - East
    - South
    - West
Same Logic, Different Data Structure

- **Stack:**
  - Push Start
Same Logic, Different Data Structure

- **Stack:**
  - **Push**
Same Logic, Different Data Structure

- **Stack:**
  - Push
Same Logic, Different Data Structure

- **Stack:**
  - **Push**
Same Logic, Different Data Structure

- **Stack:**
  - Push
Same Logic, Different Data Structure

- **Stack:**
  - Push
Same Logic, Different Data Structure

- **Stack:**
  - Push

```plaintext
<table>
<thead>
<tr>
<th>Push</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>
```
Same Logic, Different Data Structure

- **Stack:**
  - Push
Same Logic, Different Data Structure

- **Stack:**
  - Pop
Stack:
- Pop
Same Logic, Different Data Structure

- **Stack:**
  - Pop
Same Logic, Different Data Structure

- **Stack:**
  - Pop
Same Logic, Different Data Structure

- **Stack:**
  - Pop
Same Logic, Different Data Structure

- **Stack:**
  - Pop
Same Logic, Different Data Structure

- **Stack:**
  - Pop
Same Logic, Different Data Structure

- Stack:
  - Push
Same Logic, Different Data Structure

- **Stack:**
  - Push
Same Logic, Different Data Structure

- **Stack:**
  - Push Finish!
Same Logic, Different Data Structure

- **Queue:**
  - Enqueue Start
Same Logic, Different Data Structure

- **Queue:**
  - Enqueue
Same Logic, Different Data Structure

- **Queue:**
  - Enqueue
Same Logic, Different Data Structure

- **Queue:**
  - Enqueue
Same Logic, Different Data Structure

- **Queue:**
  - Enqueue
Same Logic, Different Data Structure

- **Queue:**
  - Enqueue

```
[1, 2, 3, 4, 1, 2, 3, 4, 4, 3, 2, 1, 1]
```
Same Logic, Different Data Structure

- **Queue:**
  - **Enqueue**
Same Logic, Different Data Structure

**Queue:**
- Enqueue
Same Logic, Different Data Structure

- Queue:
  - Dequeue
Same Logic, Different Data Structure

- **Queue:**
  - Enqueue
Same Logic, Different Data Structure

- Queue:
  - Enqueue
Same Logic, Different Data Structure

- Queue:
  - Enqueue Finish!

```
3 2 2 1 2 3 4 4 4 4 4 4 4 3 2 1
```

```
4 |
3 |
2 |
1 |
```

```
4,1
4,2
4,3
4,4
3,4
2,4
1,4
```

```
3,1
2,1
2,2
3,2
```
Inheritance

- One class extends another
  - Parent class: shared attributes/methods
  - Child class: more specific attributes/methods
    - Class declared extends the parent class

- Why? Lets you share code
  - Repeated code is evil
Abstraction/Inheritance Hierarchy

- More abstract
- More concrete
A. Abstract Classes

- Classes at a higher level in the hierarchy are more abstract
  - Makes no sense to instantiate them
  - For example, what is an animal, what does it look like?
    - There will be attributes that can be tracked at higher levels, but often behaviors belong at lower levels.
- The more concrete the class becomes (lower in the hierarchy), the more it can be described
A. Abstract Classes

- Back to stacks and queues
  - Both are abstract data types
  - Both have similar operations, though they are carried out differently
    - Add/push/enqueue
    - Remove/pop/dequeue
  - Might make sense to define a parent Abstract Data Type class to house both
    - But, we wouldn’t want to instantiate this class, since we don’t know what behavior it should exhibit
B. Generic Types

- Next problem with stacks and queues is that we need a different one to store different data types
  - StackOfStrings, StackOfPositions
  - QueueOfStrings, QueueOfPositions

- Really want to have a single structure that can store any type of data
  - Python is fairly lenient on this
    - Only the pieces of code that are different need to have special checks in them
  - Other, strongly typed, languages are not as forgiving
    - Languages like Java and C++ will use generic data type identifiers
C. Multiple Inheritance
C. Multiple Inheritance: The Diamond Problem
Summary

I. Stacks and Queues – walkthrough

II. Inheritance Revisited
   A. Abstract Classes
   B. Generic Data Types
   C. Multiple Inheritance