Review Topics

- unit testing
  - practical
- Maps & String Methods
- When to use abstract V.S. interface ✓
- inheritance broadly
  polymorphism ✓
- method declarations
- this keyword
- has-A v.s. Is-A ✓

OOP

has-A relationships are talking about properties

Is-A relationships refer to a class's identity

- is-a relationships are used for polymorphism
  - I can substitute a value of a child class where I want a superclass
  - I can substitute any class that implements an interface in code that expects the interface type

---

Class vs Object

- Class
  - blueprint "for an object"
    - extends for inheriting
    - implements for interfaces

```
public class Foo extends Bar implements Baz {
    public Foo(...) {
    }
}
```

```
1      3   2
Foo myFoo = new Foo(...);
```

1 allocates a variable that can hold

a reference to a Foo

2. Allocates heap space, calls the foo constructor makes a new Foo object from the class
3. assign the reference to the created object into the variable

Polymorphism (class-based) (inheritance)

```
Bar myBar = new Foo(...);
```

Polymorphism (interfaces)

```
Baz myBaz = new Foo(...);
```

Requirements to inherit

- parent cannot be a final class
- we must satisfy a constructor on the parent via super(...)
- if I am not abstract I must override any abstract methods on my parent

Requirements to implement an interface

- I must override all methods provided by the interface w/ public methods on my class

---

this

The this keyword is used when defining a class to refer to the properties and methods of the current instance once the class is instantiated into an object.

use this anywhere there might be ambiguity between a variable on the object proper and a variable defined locally in a method.

⇒ NOTE !! JavaScript also has a this keyword, but it's not

this keyword, its futility
different!