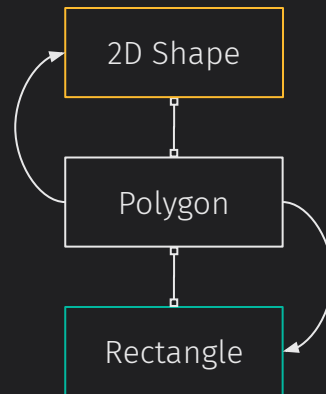


# Casting

CS 272 Software Development

# Casting

- **Upcasting** (or **implicit** casting)
  - References an object as its superclass
  - Only access methods in the superclasses
- **Downcasting** (or **explicit** casting)
  - References an object as its subclass
  - Allows access to methods in the subclass



# Casting

- Does not change the **type** of object, only changes the **reference** (or identifier) to an object
- With **overridden** methods, will call the method associated with the object type (not the reference)
- Can use casting to create **generalized** methods that work on multiple subclasses



**Module** java.base

**Package** java.lang

## Class Double

```
java.lang.Object
  java.lang.Number
    java.lang.Double
```

### All Implemented Interfaces:

Serializable, Comparable<Double>, Constable, ConstantDesc

---

```
public final class Double
  extends Number
  implements Comparable<Double>, Constable, ConstantDesc
```

The Double class wraps a value of the primitive type double in an object. An object of type Double contains a single field whose type is double.

<https://www.cs.usfca.edu/~cs272/javadoc/api/java.base/java/lang/Double.html>



# Casting Syntax

1. `// Upcasting Examples`
2. `Number n = Double.valueOf(3.14);`
3. `Object o = n;`
- 4.
5. `// Downcasting Example`
6. `Double d = (Double) n;`



# Explicit Casting

1. `// Throws a ClassCastException`
2. `Object a = new StringBuilder("3.14");`
3. `Double b = (Double) a;`



# Questions?

