BEFORE WE START

Talk to your neighbors:

What is your most-used emoji?

Instructors

Tristan Huber & Hunter Schafer

TAs

Ambika
Andrew
Audrey
Autumn
Ayush
Ben
Colton
Di
Eesha
Elizabeth

Evelyn
Jay
Jaylyn
Jin
Joe
Kevin
Leon
Megana
Melissa
Mia

Poojitha
Rishi
Rucha
Shivani
Shreya
Steven
Suhani
Yijia
Ziao

Questions during Class?
Raise hand or send here

sli.do    #cse122
Lecture Outline

• Announcements
• Interfaces Review
• More Shapes!
• Comparable
Announcements

- C2 due tomorrow (Thurs, May 18)
- P3 will be released on Fri, May 19
- Quiz 2 next Tuesday (May 23)
- Reminder that the final exam is scheduled for Tuesday (June 6) 2:30pm-4:20pm
Lecture Outline

• Announcements

• Interfaces Review

• More Shapes!

• Comparable
Interfaces - define a set of behavior which classes can implement

... like a “certification”, eg ArrayList is “certified” as a List because it can do all List behaviors.

note: interfaces say nothing about a class’ state
(PCM) List Interface

List is an interface – defines the *behaviors* which make something a List, inc:

- add, clear, contains, get, isEmpty, size

Any class with these behaviors can implement List

List documentation enumerates the full list of methods required to be a List:
https://docs.oracle.com/javase/8/docs/api/java/util/List.html
Why interfaces?

**Abstraction**

Interfaces support *abstraction* (the separation of ideas from details)
(PCM) Why interfaces?

Car interface
- Press gas pedal
- Press brake pedal
- Turn steering wheel
(PCM) Why interfaces?

**Flexibility**

public static void driveToWork(Car c) {...}

This method does not need to change if we update our implementation of Car.
Lecture Outline

• Announcements

• Interfaces Review

• More Shapes!

• Comparable
Classes can Implement Multiple Interfaces

A class can implement multiple interfaces – must include all *behaviors* from each interface it implements.

- **Battery Interface**
  - Charge
  - Discharge

- **Car Interface**
  - Press gas pedal
  - Press brake pedal
  - Turn steering wheel
Classes can Implement Multiple Interfaces

```java
public class Leaf implements Car, Battery {
    ...
}
```

But Leaf would have to implement:
- `pushGasPedal`, etc from `Car`
  
  AND
- `charge`, `discharge` from `Battery`
An interface can extend another

You can have one interface extend another

So if `public interface A extends B`, then to implement A a class needs to have all methods from A and B.

In the above example, A is more specific than B
An interface can extend another

We can write another interface

\texttt{Polygon} that extends \texttt{Shape}

- Square is a \texttt{Polygon} (and \texttt{Shape})
- Triangle is a \texttt{Polygon} (and \texttt{Shape})
- Circle is a \texttt{Shape} (but \textit{not} a \texttt{Polygon})
Select all of the following statements that would cause an error.

A) B foo = new Two();
    foo.b();

B) D bar = new Two();
    bar.a();

C) D baz = new Three();
    baz.a();

D) A waldo = new Three();
    waldo.b();
Select all of the following statements that would cause an error.

```java
public interface A {
    public void a();
}

public interface B extends A {
    public void b();
}

public interface C {
    public void c();
}

public interface D extends A {
    public void d();
    public void e();
}

public class One implements A {
    ...
}

public class Two implements B, D {
    ...
}

public class Three implements B, C {
    ...
}

public class One implements A {
    ...
}

public class Two implements B, D {
    ...
}

public class Three implements B, C {
    ...
}

A) B foo = new Two();
   foo.b();

B) D bar = new Two();
   bar.a();

C) D baz = new Three();
   baz.a();

D) A waldo = new Three();
   waldo.b();
```
Select all of the following statements that would cause an error.

public interface A {
    public void a();
}

public interface B extends A {
    public void b();
}

public interface C {
    public void c();
}

public interface D extends A {
    public void d();
    public void e();
}
Select all of the following statements that would cause an error.

```java
public class One implements A {
    ...
}
public class Two implements B, D {
    ...
}
public class Three implements B, C {
    ...
}
```

### Diagram

- **Interface A**
  - `a()`

- **Interface B**
  - `b()`

- **Interface C**
  - `c()`

- **Interface D**
  - `d()`
  - `e()`

- **Class One**
  - `a()`

- **Class Two**
  - `b()`
  - `a()`
  - `d()`
  - `e()`

- **Class Three**
  - `b()`
  - `a()`
  - `c()`

Errors:
- `public class One implements A {` (incorrect syntax, should end with `}`)
- `public class Two implements B, D {` (incorrect syntax, should not use `extends`)
- `public class Three implements B, C {` (incorrect syntax, should not use `extends`
Select all of the following statements that would cause an error.

A) B foo = new Two();
   foo.b();

B) D bar = new Two();
   bar.a();

C) D baz = new Three();
   baz.a();

D) A waldo = new Three();
   waldo.b();
Lecture Outline

• Announcements
• Interfaces Review
• More Shapes!

• Comparable
Recall the Student / Course Example from Wed

Course stored a field

private List<Student> roster;

We also had a suggestion to use a Set to store the students…

Seems like a great idea (no duplicates, not worried about keeping a specific order or indexing into it) but … Java reasons

• HashSet won’t work because of the hashCode() business I mentioned on Wed
• TreeSet won’t work because what does it mean to “sort” Students
Comparable

TreeSet uses an *interface* called `Comparable<E>` to know how to sort its elements.

Only has one required method:
```java
public int compareTo(E other)
```

Its return value is:
- `< 0` if this is “less than” other
- `0` if this is equal to other
- `> 0` if this is “greater than” other

Comparable documentation:
https://docs.oracle.com/javase/8/docs/api/java/lang/Comparable.html