

CS 152

Computer Programming Fundamentals Interfaces

Brooke Chenoweth

University of New Mexico

Spring 2025

What is an interface?

- A contract between a class and the outside world.
- A class that implements an interface promises to provide the behavior published by that interface.
- Similar to a class, contains
 - constant definitions
 - method signatures
 - *not method bodies!*
- Interfaces cannot be instantiated.
- Can be used as a type.

Example Interface

```
public interface MyInterface {  
  
    // Constants in an interface are  
    // implicitly public static and final  
    double MY_NUMBER = 17.01;  
  
    // Method signatures (end with semicolon, no body)  
    // all methods implicitly public  
    void doStuff(int i, double x);  
    int doOtherStuff(String s);  
}
```

Implementing an interface

- Use `implements` in class definition.
- A class can implement more than one interface. Separate interface names with comma.
- Class must implement all the methods declared in the interface.

Implementing interface example

```
public class MyImpl implements MyInterface {  
  
    public void doStuff(int i, double x) {  
        // I'm just going to do nothing, so there!  
    }  
  
    public int doOtherStuff(String s) {  
        return s.length();  
    }  
}
```

Implementing multiple interfaces

```
public interface Fruit {  
    boolean hasPeel();  
}
```

```
public interface Vegetable {  
    boolean isRoot();  
}
```

```
public class Tomato implements Fruit, Vegetable {  
    public boolean hasPeel() {  
        return true;  
    }  
    public boolean isRoot() {  
        return false;  
    }  
}
```

Interface as Type

```
public static void main(String[] args) {  
    Tomato t1 = new Tomato();  
    Fruit t2 = new Tomato();  
    Vegetable t3 = new Tomato();  
  
    boolean testPeel1 = t1.hasPeel();  
  
    boolean testPeel2 = t2.hasPeel();  
  
    // boolean badTestPeel3 = t3.hasPeel();  
    boolean testPeel3 = ((Tomato)t3).hasPeel();  
}
```

Interface as Type

```
public interface Relatable {  
    boolean isLargerThan(Relatable other);  
}
```

```
public static Relatable findLargest(Relatable[] items) {  
    Relatable largest = null;  
    for(Relatable item : items) {  
        if(largest == null || item.isLargerThan(largest)) {  
            largest = item;  
        }  
    }  
    return largest;  
}
```


Interface as API

- Company makes interface public to customers.
- Implementation of interface kept secret.
- Implementation may change later. Fine as long as still implements original interface.
- Sometimes customer implements the company's interface.
 - Java API

Example interfaces

- Comparable – Imposes a natural ordering of objects implementing it
- Collection – Common interface for all collections (implementing classes: ArrayList, LinkedList, Vector, etc...)
- Iterator – Provides a serialization of collections
- List – Defines methods common for lists
- etc...