

INTERFACES

CS302 – Introduction to Programming
University of Wisconsin – Madison
Lecture 30

By Matthew Bernstein – matthewb@cs.wisc.edu

What is an Interface?

- According to the *New Oxford American Dictionary*, an Interface is defined as:

“The point where two systems, subjects, organizations, etc., meet and interact”

- When you use any system, you interact with it using its interface.

- Example:

Oven: knobs, timer...

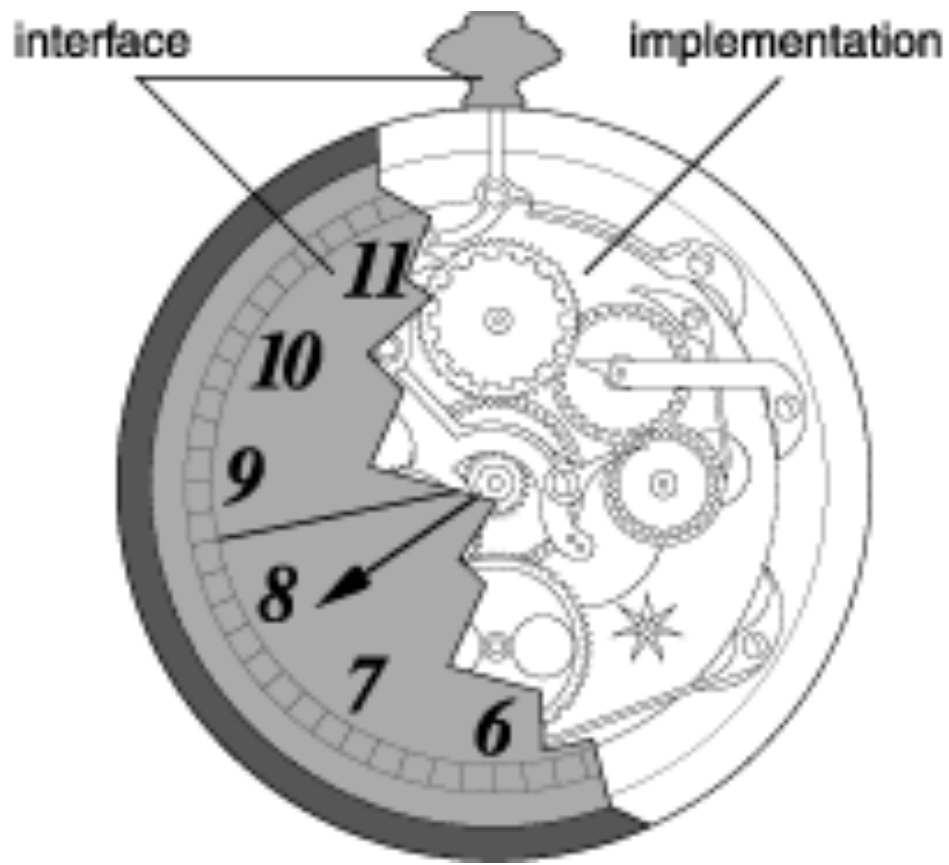
Car: steering wheel, accelerator, brake...

Laptop: keyboard, screen, mouse...

Interface vs. Implementation

- By knowing how to operate the interface, you do not need to understand the underlying implementation, machinery, and complexity working behind the interface
- Two objects might be implemented completely differently, but they may be used using the same interface
- For example, cars made by two different manufacturers may have very different implementations (different designs, electronics, engine, transmission), but they all generally use the same interface
- If this were not the case, you would have to learn how to drive all over again whenever you bought a new car

Example: a clock



Interfaces in Java

- In Java, an **interface** is a collection of method headers without any implementation.
- An interface does NOT have any code
- If a class implements a specific interface, then that class **MUST** implement all of the methods in that interface
- Each interface is written in its own .java file of the same name

Example Interface

The “interface” reserved word denotes this is an interface, not a class

Interfaces are named using the same standards as classes: camel-case with the first letter capitalized

```
public interface Measurable
{
    public double getMeasure();
}
```

No statements are specified in the interface. The interface simply describes the methods.

This would all be in a file called Measurable.java

Implementing an Interface


- You can have your class **implement** an interface
- By implementing an interface, you are “signing a contract” that this class implements all of the methods in the interface
- Example:

```
public class BankAccount implements Measureable
{
    private double balance;

    ...

    public double getMeasure()
    {
        return this.balance;
    }

    ...
}
```



The reserved word **implements** denotes that this class implements an interface

Interface Parameters

- You can write a method that accepts any object that implements a specific interface
- Example:

This method accepts
any object that implements
the `Measureable` interface



```
public boolean isValidMeasure(Measurable m)
{
    // ... Do Stuff
}
```


Implementing Multiple Interfaces

- Your class can implement multiple interfaces
- Example:

```
public class MyClass implements Interface_A, Interface_B
{
    // Must implement all methods in all interfaces
}
```

Notes on Interfaces

- If your class implements an interface, you **MUST** implement all of the methods in that interface
- If you do not, then you will have a compile-time error