4. Interfaces :: Java Interface Rules :: Part II

1. If an interface is not a class, then how do we use it? An interface is **implemented** by classes and an interface may be implemented by multiple classes:

```
// SomeInterface.java
public interface SomeInterface {
  void abstractMethod1();
}
// SomeClass.java
public class SomeClass implements SomeInterface {
  public void abstractMethod1() { ... }
}
// SomeOtherClass.java
public class SomeOtherClass implements SomeInterface {
  public void abstractMethod1() { ... }
}
```

Note that even though *abstractMethod1*() is **public** by default in *SomeInterface*, we must specify an accessibility modifier of **public** in *SomeClass* and *SomeOtherClass* when implementing the abstract method.

4. Interfaces :: Java Interface Rules :: Part II (continued)

2. A class that implements an interface must implement all of the abstract methods declared in the interface.

```
// SomeInterface.java
public interface SomeInterface {
  void abstractMethod1();
  void abstractMethod2();
}

// SomeClass.java
public class SomeClass implements SomeInterface {
  public void abstractMethod1() { ... }
  public void abstractMethod2() { ... }
}

// SomeOtherClass.java
public class SomeOtherClass implements SomeInterface {
  public void abstractMethod1() { ... }
  public void abstractMethod2() { ... }
}
```

4. Interfaces :: Java Interface Rules :: Part II (continued)

3. A class may implement multiple interfaces.

```
// InterfaceA.java
public interface InterfaceA {
  void fribble();
  void gribble();
}

// InterfaceB.java
public interface InterfaceB {
  void blorp();
  void glorp();
}
// SomeClass.java
public class SomeClass implements InterfaceA, Interface B {
  public void fribble() { ... }      // Must implement InterfaceA.fribble()
  public void gribble() { ... }      // Must implement InterfaceA.gribble()
  public void blorp() { ... }        // Must implement InterfaceB.blorp()
  public void glorp() { ... }        // Must implement InterfaceB.glorp()
  public void otherMethod() { ... }
}
```