5. Interfaces :: Java Interface Rules :: Part III

1. When an interface type is used to declare an object variable then the only methods that may be called on that object variable are the interface-implemented methods :

```
public interface IFace {
  void fribble();
  void gribble();
}

public class SomeClass implements IFace {
  public SomeClass() { ... }       // A public ctor.
  public void fribble() { ... }    // Implements IFace.fribble().
  public void gribble() { ... }    // Implements IFace.gribble().
  public void foo() { ... }        // An instance method.
  public void bar() { ... }        // Another instance method.
}

public void someMethod(IFace pObj) {
  pObj.fribble();  // Legal
  pObj.gribble();  // Legal
  pObj.foo();      // Illegal
  pObj.bar();      // Illegal
}
```

5. Interfaces :: Java Interface Rules :: Part III (continued)

2. A class may both extend a superclass and implement one or more interfaces:

```
public interface Drawable {
  ...
}
public interface Serializable {
  ...
}
public class Square extends Rectangle implements Drawable, Serializable {
  ...
}
```

5. Interfaces :: Java Interface Rules :: Part III (continued)

3. An abstract class my implement an interface but is not required to actually implement the interface methods—although subclasses *are*.

```
public interface Drawable {
  void draw(Graphics gc);
}

public abstract class Shape implements Drawable {
  // Shape does not have to provide an implementation of Drawable.draw().
}

public class Rectangle extends Shape {
  // Rectangle must provide an implementation of Drawable.draw().
  public void draw(Graphics gc) {
    // Rectangle provides code to draw the rectangle.
  }
}
```

Note that it is not necessary to specify that *Rectangle* implements *Drawable* because subclasses inherit interfaces implemented by superclasses. That said, it does not hurt to write:

```
public class Rectangle extends Shape implements Drawable { ... }
```

There are additional Java rules regarding interfaces but the ones we have discussed will be sufficient for this course.