

Lecture 5 Clicker Questions

1) Which line of this program is incorrect?

```
A. public interface Colorable {  
    public Color getColor() {  
        B. return Color.WHITE;  
    }  
}
```

```
C. public class Rectangle implements  
Colorable {  
    //constructor elided  
    D. @Override  
    public Color getColor() {  
        E. return Color.PURPLE;  
    }  
}
```

Answer: B. The interface is just a contract. It tells implementing classes *what* methods to define - not *how* to define them.

2) Given the following interface:

```
public interface Clickable {  
    public void click();  
}
```

Which of the following would work as an implementation of the `Clickable` interface? (don't worry about what `changeXPosition()` does)

<pre>A @Override public void click() { this.changeXPosition(100.0); }</pre>	<pre>C @Override public void clickIt() { this.changeXPosition(100.0); }</pre>
<pre>B @Override public void click(double xPosition) { this.changeXPosition(xPosition); }</pre>	<pre>D @Override public double click() { return this.changeXPosition(100.0); }</pre>

Answer: A. The implementing class needs to implement the method with the same exact signature as the interface. Same return type, name (case sensitive!), and parameters.

3) Given the following class:

Given that `typeable` has declared the `type()` method and `clickable` has declared the `click()` method, which of the following calls is valid?

A <code>Typeable macBook= new Typeable();</code> <code>macBook.type();</code>	C <code>Typable macBook= new Laptop();</code> <code>macBook.click();</code>
B <code>Clickable macBook = new Clickable();</code> <code>macBook.type();</code>	D <code>Clickable macBook = new Laptop();</code> <code>macBook.click();</code>

Answer: D. You can't instantiate an instance interface - you must instantiate an instance of a class that implements that interface. This would eliminate A and B. D is the only option that calls a method defined by the declared type.