What gets printed when the following program is run?

```java
public class While {
    public static void main( String[] args ) {
        final int MAX = 5;
        int i = 3, j = -1;
        while ( i < MAX ) {
            j = i;
            while ( j <= MAX ) {
                System.out.println( i + " " + j );
                j += 2;
            }
            i++;
        }
        System.out.println( i + " " + j );
    }
}
```

Trace the following program and specify its output.

```java
public class Trace {
    public static void main( String[] args ) {
        foo2();
        System.out.println( "main1" );
        foo3();
        System.out.println( "main2" );
        foo1();
        System.out.println( "main3" );
    }
    public static void foo1() {
        System.out.println( "A" );
    }
    public static void foo2() {
        System.out.println( "B" );
        foo1();
    }
    public static void foo3() {
        foo2();
        System.out.println( "C" );
    }
}
```

An object which is defined to handle action events must implement the _________________ interface and provide a body/implementation for the abstract method named _________________.

A GUI _________________ has a layout manager which defines how GUI _________________ are laid out.
Given the following definitions:

```java
public interface Speakable
{
    public abstract String speak();
}
```

```java
class Thing1 implements Speakable
{
    private static final String SPEAK = "Me";
    public Thing1()
    {
        // ctor initialization here
    }
    public String speak()
    {
        return SPEAK;
    }
    public void doit( String s )
    {
        // Thing1 does its thing with s
    }
}
```

```java
class Thing2 implements Speakable
{
    public static final String SPEAK = "No, Me";
    public Thing2()
    {
        // ctor initialization here
    }
    public String speak()
    {
        return SPEAK;
    }
    public void doit()
    {
        // Thing2 does its thing
    }
}
```

And the following variable definitions:

```java
Thing1 thing1;
Thing2 thing2;
Speakable speakable;
```

Indicate which are valid Java statements. Consider each statement executed sequentially in the order it appears.

1) Invalid Java statement – Compiler Error
2) Valid Java statement – No Compiler Error

```
things = new Thing1();
thing1.speak();
thing1.doit();
thing1.doit( "Yours" );
String s1 = Thing1.SPEAK;
thing2 = new Thing2();
thing2.doit( "Theirs" );
thing2.speak();
thing2.doit();
```

```
String s2 = Thing2.SPEAK;
speakable = new Speakable();
thing1 = thing2;
thing1 = speakable;
speakable = new Thing1();
speakable.speak();
speakable.doit( "Mine" );
speakable = thing2;
speakable.speak();
speakable.doit();
```

**Hint:** What does the compiler know about any reference variable at compile time (vs. run time)?