1. Given the following class definitions:

```java
public class Animal {
    public void method1() {
        System.out.println("Animal 1");
    }
    public void method3() {
        System.out.println("Animal 3");
    }
}
public class Reptile extends Animal {
    public void method1() {
        System.out.println("Reptile 1");
    }
    public void method2() {
        System.out.println("Reptile 2");
    }
}
public class Mammal extends Animal {
    public void method2() {
        method3();
        System.out.println("Mammal 2");
    }
    public void method3() {
        System.out.println("Mammal 3");
    }
}
public class Dog extends Mammal {
    public void method1() {
        System.out.println("Dog 1");
    }
    public void method3() {
        method1();
        super.method3();
        System.out.println("Dog 3");
    }
}
```

What is the output of the following code:

```java
Animal var1 = new Dog();
var1.method3();
```

Put your answer here:
2. Fill in each blank with one of the following key words: abstract, throws, extends, new, implements, interface.

The Java keyword which denotes inheritance of interface is ___________________.

The Java keyword which denotes that a class definition is incomplete is _______________.

The Java keyword which denotes inheritance of implementation is _____________________.

In Java, a(n) _______________ can only declare public abstract methods and public static final constants while a(n) _______________ class can have a mixture of concrete and abstract methods and instance variables.

3. Given the following class definitions for class Bar, class Fubar, and class FubarTest:

```
public class Bar {
    public Bar( int x, int y ) {
        System.out.println( "Bar ctor #1" );
    }

    public Bar() {
        this(2, 5);
        System.out.println( "Bar ctor #2" );
    }

    public String toString() {
        System.out.println( "Bar.toString" );
        return "Bar.toString";
    }
}
```

```
public class Fubar extends Bar {
    public Fubar() {
        super();
        System.out.println( "Fubar ctor #1" );
    }

    public Fubar( int x, int y ) {
        super( x, y );
        System.out.println( "Fubar ctor #2" );
    }

    public String toString() {
        System.out.println( "Fubar.toString" );
        return "Fubar + " + super.toString();
    }
}
```

```
public class FubarTest {
    public static void main( String[] args ) {
        Bar ref = new Fubar();
        System.out.println( "-----" );
        System.out.println( ref.toString() );
    }
}
```

What is the output when we run FubarTest as:

```
java FubarTest
```