**INTERFACES IN JAVA**

Example of interface implementation:

Interface methods are, by default, **abstract** (no body, have to be overridden) and **public**.

**interface Animal {**

**public void animalSound();**

**public void run();**

**}**

Interfaces are used using the keywork **implements**.

When a class implements an interface, it must implement all the methods of the interface. (Sort of like a contract)

**class Pig implements Animal {**

**public void animalSound() {**

**System.out.println(“Pig sound”);**

**}**

**public void run() {**

**System.out.println(“Pig is running”);**

**}**

**}**

**class MainClass {**

**public static void main(String[] args) {**

**Pig firstPig = new Pig();**

**firstPig.animalSound();**

**}**

**}**

That is all the basic stuff. But with Java 8 there is more to interfaces:

Interfaces in Java can now have **default** and **static** methods.

Example of a default method in an interface:

public interface Interface1 {

void method1(String str);

default void log(String str) {

System.out.println(“I1 logging: “ + str);

}

}

When a class implements an interface, it is not mandatory to provide implementation for default methods of an interface.

Example of a static method in an interface:

public interface MyData {

static boolean isNull(String str) {

if(str == null) return true;

else return false;

}

}

Java interface static methods are similar to the default method except that we cant override them in the implementation classes.

Functional interfaces: covered in lambda expressions!