**Java 8 Functional Interface**

An interface with exactly one abstract method is called Functional Interface. @FunctionalInterfaceannotation is added so that we can mark an interface as functional interface.

All functional interfaces are recommended to have an informative @FunctionalInterface annotation. This not only clearly communicates the purpose of this interface, but also allows a compiler to generate an error if the annotated interface does not satisfy the conditions.

**Any interface with a SAM(Single Abstract Method) is a functional interface**, and its implementation may be treated as lambda expressions.

Note that Java 8’s default methods are not abstract and do not count: a functional interface may still have multiple default methods.

@FunctionalInterface annotation is useful for compilation time checking of your code. You cannot have more than one method besides static, default and abstract methods that override methods in Object in your @FunctionalInterface or any other interface used as a functional interface.

But you can use lambdas without this annotation as well as you can override methods without @Override annotation.

**This can be used in lambda expression:**

public interface Foo{

public void doSomething();

}

**This cannot be used in lambda expression:**

public interface Foo{

public void doSomething();

public void doSomethingElse();

}

**But this will give compilation error:**

@FunctionalInterface

public interface Foo{

public void doSomething();

public void doSomethingElse();

}

## **Java 8 Programming Language Enhancements**

Java 8 provides following features for Java Programming:

* Lambda expressions,
* Method references,
* Functional interfaces,
* Stream API,
* Default methods,
* Base64 Encode Decode,
* Static methods in interface,
* Optional class,
* Collectors class,
* ForEach() method,
* Parallel array sorting,
* Nashorn JavaScript Engine,
* Parallel Array Sorting,
* Type and Repating Annotations,
* IO Enhancements,
* Concurrency Enhancements,
* JDBC Enhancements etc.

#### Q=what is **StampedLock in java 8?**