Java 8 Features

Below are the some of the Important features of Java 8

1) **forEach() method in Iterable interface.**

```
    List<Employee>empList = new LinkedList<>();
// ... add some emp objects to the collection
    empList.forEach(emp -> System.out.println(emp));
```

2) **default and static methods in Interfaces.**

```
interface Java8Example {

public defaultvoid defaultMethod() {
        System.out.println("This is Default Method In Interface");
}

public staticvoid staticMethod() {
        System.out.println("This is static method defined in Interface");
}
}
```

3) **Functional Interfaces**

- An Interface that contains exactly one abstract method is known as functional interface
- Functional Interface is also known as Single Abstract Method Interfaces
  - It can have any number of default, static methods but can contain only one abstract method

```
@FunctionalInterface
interface Java8Example {

    publicdefaultvoid defaultMethod() {
            System.out.println("This is Default Method In Interface");
    }

    publicstaticvoid staticMethod() {
            System.out.println("This is static method defined in
Interface");
    }
    publicvoidsingleAbstractMethod();
}
```

4) **Lambda Expressions.**

- The Lambda expression is used to provide the implementation of an interface which has functional interface. It saves a lot of code

```
Runnable r1 = new Runnable() {
@Override
publicvoid run() {
System.out.println("This is the Implementation without using Lamda Expression");

        }
};

//By considering Lamda Expression in Java 8
```

Runnable r2 = ()-> System.***out***.println("This is the implementation by considering the Lamda Expression");

5) **Java Stream API for Bulk Data Operations on Collections.**
6) **Method References**
   - Reference to a static method

   (args) -> Class.staticMethod(args) can be write as **Class::staticMethod**

   - Reference to an instance method. - **They can only be used to replace a single-method lambda expression.**

(args) -> obj.instanceMethod(args) can be written as obj::instanceMethod

   - Reference to a constructor

(args) -> new ClassName(args)can be written asClassName::new

7) **Java Time API**

   There are few issues with existing Date and Time API such as thread safety , The *Date* and *Calendar* APIs are poorly designed, ***ZonedDate* and *Time developers has to write additional logics***

   - *Java 8 provides few APIs such as Using* LocalDate*,* LocalTime *and* LocalDateTime

   LocalDate localDate = LocalDate.now();LocalTime now = LocalTime.now();

8) **Collection API improvements.**
   - *Performance Improvement for HashMaps with Key Collisions*
   - *Lamda expressions, Default Methods, streams, java.util.function/stream*

9) **Concurrency API improvements.**
   - New methods in java.util.concurrent.ConcurrentHashMap(include various forEach methods (forEach, forEachKey, forEachValue, and forEachEntry), search methods (search, searchKeys, searchValues, and searchEntries) and a large number of reduction methods (reduce, reduceToDouble, reduceToLong etc.)
   - New classes in java.util.concurrent.atomic(set of new classes (DoubleAccumulator, DoubleAdder, LongAccumulator, LongAdder)
   - New methods in java.util.concurrent.ForkJoinPool(Two new methods (getCommonPoolParallelism() and commonPool()) have been added)
   - New class java.util.concurrent.locks.StampedLock - new StampedLock class adds a capability-based lock with three modes for controlling read/write access (writing, reading, and optimistic reading

10) Nashorn JavaScript Engine

- jjs command-line tool as well as using Oracle Nashorn as an embedded scripting engine inside Java applications
- It shows the Java-to-JavaScript interoperability
- This provides seem less integration between Java and Java Script

11) **jdeps command-line tool** allows the developer to analyze class files to determine package-level or class-level dependencies**.**

12) **Java Optional Class:**A container object which may or may not contain a non-null value. If a value is present, isPresent() will return true and get() will return the value