Java 11 Features :

Modularity : The **Modularity** feature is not available in Java 8. It is one of the biggest changes that have been added in Java 9. By using modularity, we can split our programs or applications into different modules. Modularity allows us to test our application easily and also managing access to what parts of our code are public-facing. We can compile our code without compiling the file by simply placed it into the module having no name. It behaves in the same way as the Java 8 compile code.

Release Flag :

### --release flag

It is a very simple combination of **-target** and **-source** that allows us **-bootclasspath** for compiling at the older version. The **-release 8** is an example that we need to compile at a Java 8 level.

* **Multi-jar** provides a way to still the code with the latest features of Java 11 while maintaining support and functionality for Java 8 clients.
* Soe new String Methods are Launched in Java 11..

Several new methods of String such as isBlank(), lines(),repeat(n), stripLeading(), stripTrailing(), and strip() are introduce.

* Java 11 allows us to use var variables to be used in lambda expressions.
* public class VarInLambdaExample {
* public static void main(String[] args)
* {
* IntStream.of(1, 2, 3, 5, 6, 7)
* .filter((var i) -> i % 2 == 0)
* .forEach(System.out::println);
* }
* }
* In Java 11, there are various methods, such as writeString(), readString(), and isSameFile() which help us to perform several operations on file.
* **isSameFile():** This method is used to know whether two paths locate the same file or not.
* Pattern recognizing is possible with the help of the asMatchPredicate() method.

This method is similar to Java 8 method asPredicate(). Introduced in JDK 11, this method will create a predicate if the pattern matches with the input string.

* **Java Deployment Technologies are removed in Java 11.**
* **There were some Java EE and CORBA Modules added in Java 9 are removed in Java 11.**

#### 4. Epsilon Garbage Collector

This handles memory allocation but does not have an actual memory reclamation mechanism. Once the available Java heap is exhausted, JVM will shut down.

Its goals are:-

* Performance testing
* Memory pressure testing
* last drop latency improvements

#### Removal of thread functions

* stop(Throwable obj) and destroy() objects that have been removed from the JDK 11 because they only throw UnSupportedOperation and NoSuchMethodError respectively. Other than that, they were of no use.

#### 7. TimeUnit Conversion

This method is used to convert the given time to a unit like DAY, MONTH, YEAR, and for time too.

TimeUnit c = TimeUnit.DAYS;

**Output:** DAYS

jshell>c.convert(Duration.ofHours(24));

**Output:** 1

jshell>c.convert(Duration.ofHours(50));

**Output:** 2

**Optional.isEmpty():** This method returns true if the value of any object is null and else returns false.