**JAVA 8 FEATURES**

**Lambda expressions**

* It is an important feature of java 8
* It is mainly used to provide implementation for functional interface.
* It consists of 3 parts : argument list, arrow operator and body
* Syntax: () -> { }
* The argument/parameter list can be null or empty

**Functional Interface**

* It is an interface which contains only one abstract method
* They are represented using the annotation ‘@FunctionalInterface’. It is not mandatory to use this annotation
* This annotation ensures that the functional interface can have only one abstract method. If we try to define another method in this functional interface an error will be raised.
* But it can have any number of default and static methods
* Lambda expressions are used to provide implementation for the abstract method in functional interfaces.
* Eg: ‘Runnable’ interface has only one abstract method ‘run()’, it is a functional interface

**Method References**

* It is used to refer method of functional interfaces
* It is an easy form of lambda expression
* It is denoted using ‘::’
* Syntax to refer method: objectname::methodname
* There are 3 types of method reference,

1. Reference to a static method

Syntax: classname::staticmethodname

1. Reference to a instance method

Syntax: objectname::methodname

1. Reference to a constructor

Syntax: classname::new

**Optional**

* It is a class mainly introduced to deal with NullPointerException
* It is residing in java.util package
* It is very helpful in knowing whether a variable has null value or not

**Streams**

* A stream is a sequence of objects that supports various methods which can be pipelined to produce the desired result
* They are wrappers around a data source allowing to operate with the data data source and processing fast
* It takes input from collections, arrays etc
* It is residing in java.util package

**forEach**

* It is a method introduced to traverse the collection elements easily
* It lies in Iterable interface as a default method
* It is like for loop, but it is designed to work with the collection elements

**Default method in interface**

* Interfaces can only have abstract methods. But as of Java 8, we can declare the default methods in interface
* In this way we can add new functionality to an existing interface without affecting the classes that implemented them.
* It allows backward compatibility which is adding new features without breaking the old code.
* It is defined using ‘default’ keyword

**Static method in interface**

* They are like default methods
* We can define static methods in interfaces using ‘static’ keyword

**Collectors class**

* It is a final class that extends Object class
* It provides reduction operations and many other operations