**Day 6: 5 – Aug 2024 – Core Java**

**Java 8 Features ie Lambda expression and stream api**

1. From java 8 onward interface can contains method with body. But we need to use static or default keyword.

If we want to use lambda expression. If need interface with only one abstract methods. It can contains more than one default as well as static.

The interface which contains only one abstract method that interface is known as functional interface. We can use @FuntionalInterface annotation to verify that interface is functional interface or not.

Lambda expression method implementation return value without return keyword. if we use one line statement. If we need to write more than one line statement then we need to use curly braces and we need to return value explicitly using return keyword.

Stream API

Steam flow of data.

Using Stream API we can load the data from collection on demand. Stream can hold the data for temporary purpose. We can apply business logic on that data to verify result those result will not effect for actual container.

Collection or Array-🡪 Stream -🡪 Intermediate op -🡪Intermediate op🡪terminal op

Stream api provided more than one intermediate operator or method. These method return type is stream itself. Terminal operator return it non stream it can be void, int, float, boolean etc.

4 functional interfaces

Java 8 provided pre defined package ie function. It is a sub package of util package. Which contains top most functional interfaces.

javap java.util.function.Function

1. Function :

apply abstract method: it takes T as parameter and return R value.

1. Supplier

get abstract method. it doesn’t take any parameter but return T value.

1. Consumer

accept abstract method. it takes T parameter but no return type.

1. Predicate

test method takes T as parameter and return boolen value.

Method reference : Java 8 provided one of the great features ie method reference which hep to refer the functional interface method.

3 types

Static method reference

Non static method reference

Constructor reference