Module 9:

**Java 8 features**

1. From java8 onward interface can contains method with body. But that method must be default or static.
2. If method is default as well as the class which implements that interface no need to provide the body. But if we want to override you can override default method static method can’t override.
3. The interface static method you need to call interfaceName.methodName()
4. Functional interface: the interface contains only one abstract method. it can contain more than one default or static but only one abstract method is known as functional interface.
5. **Lambda expression :** using lambda expression from java8 onward we can achieve functional programming concept(function within another function). Lambda provide the features an **anonymous method**. using lambda we can provide the body for functional interface method without writing explicit class. it is use to write shortest method logic. If we want to use lambda concept we need functional interface. In lambda expression it return value without return keyword. if we write multi line statement you need to return mandatory with curly braces method.

Syntax

interfaceName objectReferceName = (parameterList)->statement;

1. Stream api : using stream API we can process the data from collection of classes in function style. With help stream api on demand we can load the data, apply business logic and result the result. Steam hold the data for temporary purpose.

Java stream api provided top most 4 functional interface.

**Function : public abstract R apply(T); it can any type of value and return R value.**

**Supplier : public abstract T get(); no passing parameter but return T value.**

**Predicate: public abstract boolean test(T); pass T value but return boolean**

**Consumer : public abstract void accept(T); pass T but no return type**

These all interfaces part of function package and function package is sub package of util package.

collection classes or

array of any type ----🡪 stream -🡪IO1🡪IO2-🡪Ion-🡪Terminal operator

IO means intermediate operator : that operator takes parameter as functional interface which we need to pass in lambda style and return is stream itself. And terminal operator also take parameter as function interface in lambda style but return non stream ie void, Boolean, int etc.

Method reference also another features of java 8. Which help to write more concise alternate way to lambda expression.

We can use static method reference, non static method and constructor method reference.