**Java-8 – Introduction to stream**

* **Streams:**

There are two Streams in Java.

java.io.streams

Java 1.8 Streams

IO streams is mainly used to deal with input output operation. That is if we want to read from or write to a file, then we have to make use of java.io.streams.

Java 1.8 streams concept applicable only for Collections. If you want to process something with collection data, then we have to go for streams concept.

* **Collection VS Streams:**

Collection is used represent the group of individual objects as a single entity.

Streams is used to process objects from the collection.

* **Methods of streams filter & Map:**

List<Integer> l = new ArrayList<>();

l.add(0);

l.add(10);

l.add(20);

l.add(5);

l.add(15);

l.add(25);

System.out.println(l); [0, 10, 20, 5, 15, 25]

Requirement: Get the list of even numbers.

* **Without Streams until 1.7:**

List<Integer> l1 = new ArrayList<>();

for(Integer i1: l){

if(i%2 ==0){

l1.add(i1);

}

}

System.out.println(0,10, 20]

* **With Streams (From 1.8 version onwards)**

List<Integer> l1 = l.stream().filter(I -> I % 2 ==0).collect(Collector.toList));

System.out.println(l1); [0,10,20]

* **Example:**

import java.util.List;

import java.util.ArrayList;

import java.util.stream.\*;

class Test{

public static void main(String[] args){

List<Integer> l = new ArrayList<>();

l.add(0);

l.add(10);

l.add(20);

l.add(5);

l.add(15);

l.add(25);

System.out.println(l);

List<Integer> l1 = l.stream().filter(I -> I % 2 == 0).collect(Collectors.toList());

System.out.println(l1);

}

}

* **Example\_02:**

Requirement: There are some values in the collection, I need a resulting collection with double value. Example, if the value is 7 in then it should be 14.

Without streams.

List<Integer> l1 = new ArrayList<>();

for(Integer l1 : l){

l1.add(l1 \* 2);

}

System.out.println(l1); [0, 20, 40, 10, 30, 50]

With streams:

List<Integer> l1 = l.stream().map(I -> I \* 2).collect(Collectors.toList()); [0, 20, 40, 10, 30, 50]

* **Note: This document contains contents of video 79 & 80**