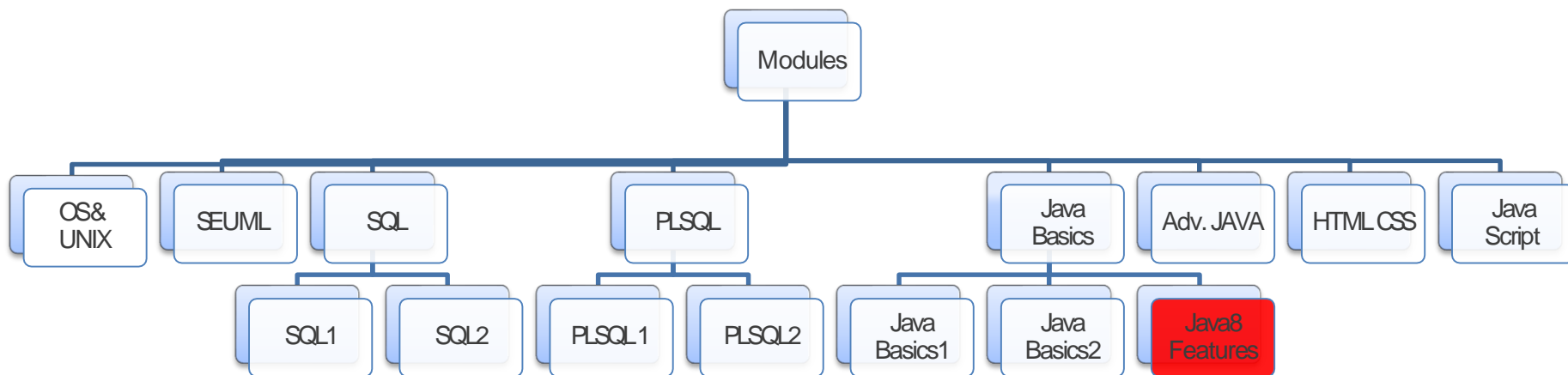


Java 8 Features

Module Overview

Purpose:

- The following modules are identified to build the basic IT skills and acquaint you with the technology basics.
- The current module (highlight in red) is contributing on Basics of Java Programming.



* **Recommended duration: 8 hours**

Module Objectives

By the end of this **module**, learner should be able to:

- Write Java programs using Lambda expressions
- Write Java programs using Stream API
- Write Java programs using Date & Time API

Lambda Expression - Use Lambda Expression

What is Lambda expression?

A lambda expression is an anonymous function that provides a concise and functional syntax, which is used to write anonymous methods.

Usage of lambda expressions

- Java lambda expressions are Java's first step into functional programming.
- In some cases it is better to use Functional programming over OOP for better readable and maintainable code.
- A Java lambda expression is thus a function which can be created without belonging to any class.
- A lambda expression can be passed around as if it was an object and executed on demand.
- Functional interfaces provide target types for lambda expressions and method references. Each functional interface has a single abstract method (SAM), called the functional method for that functional interface, to which the lambda expression's parameter and return types are matched or adapted.

```
Calculator cal = (int a, int b) -> { return a+b; }
```

```
Calculator cal = (int a, int b) -> a - b;
```

```
Calculator cal = (a, b) -> a * b;
```

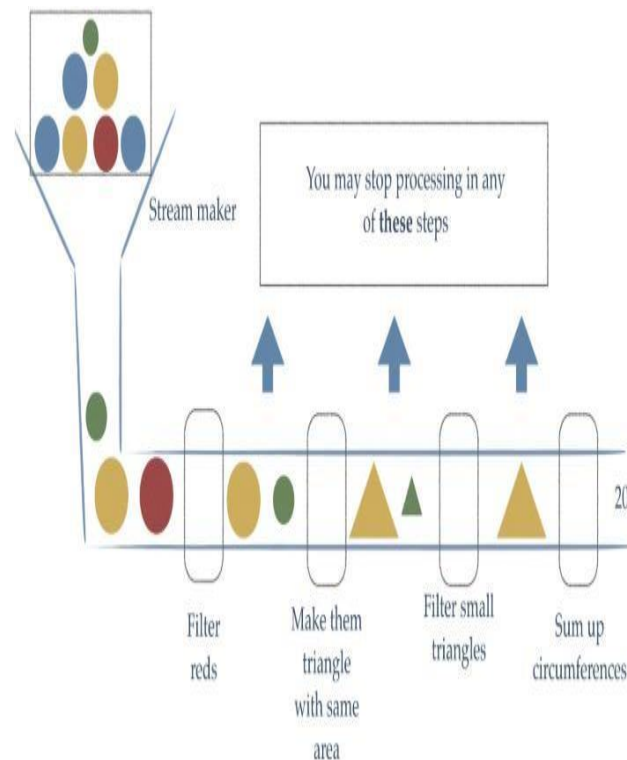
References

- <https://www.journaldev.com/2763/java-8-functional-interfaces>

Stream API – Using Stream API

What is a Stream?

- The java Stream is a data structure that is computed on-demand.
- Java Stream doesn't store data, it operates on the source data structure (collection and array) and produce pipelined data that we can use and perform specific operations. Such as we can create a stream from the list and filter it based on a condition.
- Java Stream operations use functional interfaces, that makes it a very good fit for functional programming using lambda expression.
- Java 8 Stream internal iteration principle helps in achieving lazy- seeking in some of the stream operations. For example filtering, mapping, or duplicate removal can be implemented lazily, allowing higher performance and scope for optimization.
- Java Streams are consumable, so there is no way to create a reference to stream for future usage. Since the data is on- demand, it's not possible to reuse the same stream multiple times.
- Java 8 Stream support sequential as well as parallel processing, parallel processing can be very helpful in achieving high performance for large collections.



References

- <https://www.journaldev.com/2774/java-8-stream>

Using Date and Time

What is java.time package?

- This is the base package of new Java Date Time API. All the major base classes are part of this package, such as LocalDate, Local Time, LocalDateTime, Instant, Period, Duration etc.
- All of these classes are immutable and thread safe. Most of the times, these classes will be sufficient for handling common requirements
- All the classes in the new Date Time API are immutable and good for multithreaded environments.
- The new API separates clearly between human readable date time and machine time (Unix timestamp). It defines separate classes for Date, Time, DateTime, Timestamp, Timezone etc.

References

- <https://www.journaldev.com/2800/java-8-date-localdate-localdatetime-instant>

Additional References

- The last discussed topics aimed at to gain the concept knowledge and the practical contexts to apply the knowledge.
- If you want to explore more and build the expertise level, refer to below links and books:

Links :-

<https://www.oracle.com/technetwork/java/javase/8-whats-new-2157071.html>

<https://howtodoinjava.com/java-8-tutorial/>

<https://beginnersbook.com/2017/10/java-8-features-with-examples/>

Books :-

- Beginning Java 8 Language Features
- Java 8 New Features: A Practical Heads-Up Guide

Self Check?

Instructions: You are required to document all the answers, programs and the scenarios in a word document and share it with your mentor as and when required.

- What is a functional interface?
- What is lambda expression?
- How to create Lambda expression?
- What is a Stream?
- How to create Stream in Java8?
- What is terminal operation in stream?
- What is Map operation in stream?
- What is Filter operation in stream?
- What are LocalDate and LocalTime classes?

Assignment

- Refer *Assignment Document* to complete the tasks on the required timeline
- You are required to submit the Solutions for the given assignment and refer the Participant guide to get know the submission procedure.

Module Summary

Now that you have completed this module, you will be able to:

- Explain – Lambda expressions and Stream API
- Write Java programs using Lambda expressions
- Write Java programs using Stream API
- Write Java programs using Date & Time API

Thank You

Thank you!