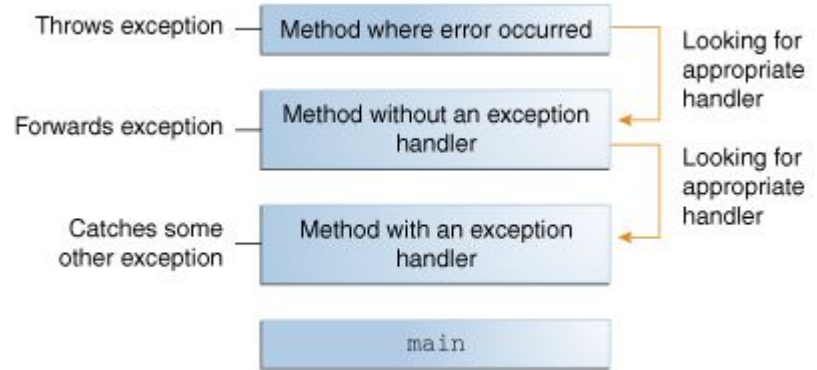
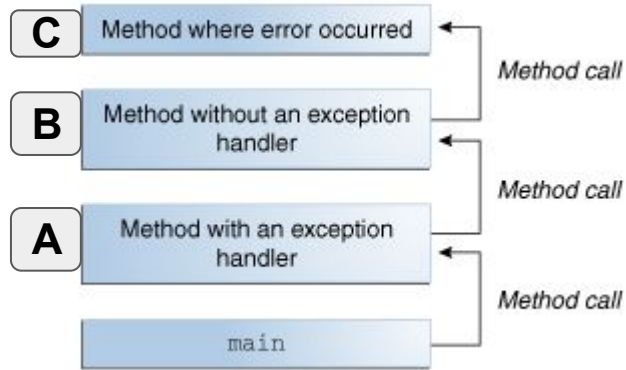


CMPT 225

Lab 04

Exceptions



Source: <https://docs.oracle.com/>

Exceptions

Three types:

- Checked Exception :

- Error

- Runtime Exception

All exceptions are checked exceptions, except for those indicated by Error and RuntimeException and their subclasses.

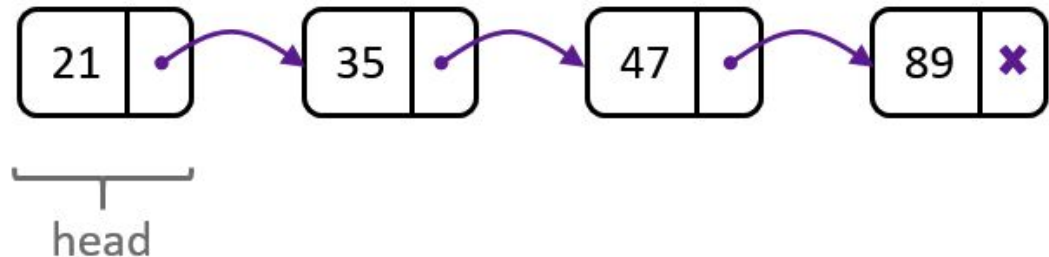
Runtime Exceptions and Errors are **not** subject to the **Catch or Specify Requirement**. They both are collectively known as **Unchecked Exceptions**.

Linked List - Recap

Chain of separate elements

Head points to the first element

When we don't know the number of items we need, we would prefer to use LinkedList instead of Array.



Source: [101computing](https://www.101computing.com/linked-lists/)

Linked List Implementation

A generic class MyLinkedList

Uses a static inner class - Node, to represent individual node inside the MyLinkedList class

This class implements the MyList interface

Add implementations to these methods :

removeAllOccurrences, reverse, merge, addBefore

(Extra: implement a recursive implementation for reverse)

Predicate

Implementing filter : we'll use predicate to implement this method

It's a functional interface* defined in `java.util.function.Predicate`

Its main methods :

```
boolean test(T t)
```

```
default Predicate<T> and(Predicate<? super T> other)
```

```
default Predicate<T> negate()
```

*a functional interface has exactly one abstract method

Lambda Expression

A short block of code -consists of following

A comma-separated list of formal parameters enclosed in parentheses

The Arrow token ->

A body, which consists of a single expression or a statement block

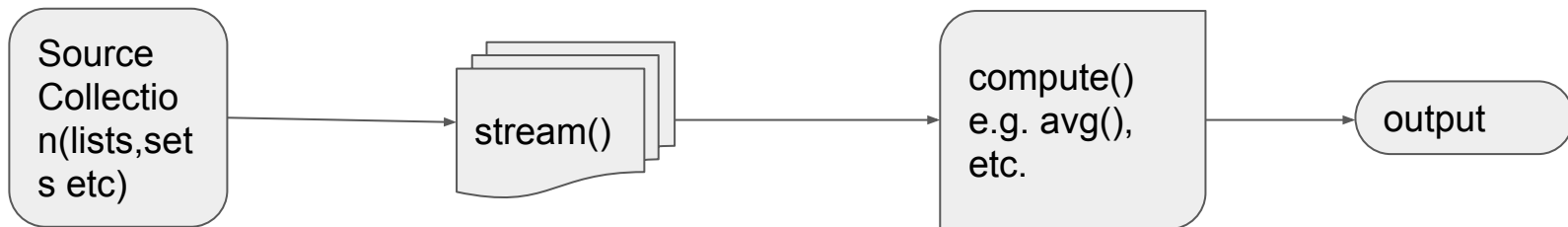
e.g.

(p) -> p>5

Lambda expressions are objects.

Stream

Defined in `java.util.stream`

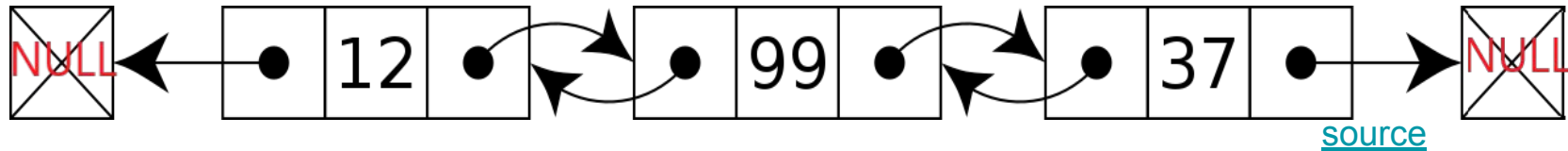


A sequence of elements supporting sequential and parallel aggregate operations([source](#)).

The source collection is not changed.

Doubly Linked List

Each node has two references - Next and Prev



Some advantage and usage:

- Fast Deletion of a given node since we have references to both next and prev

- Traversal in both direction possible.

- Used in browsers to implement backward and forward pages

Doubly Linked List

Implement all the missing methods:

size, addToHead, addToTail, removeFromHead, removeFromTail,
add, isEmpty, toString, reverse

(Here we need to change both prev and next references, and also need to keep updating head and tail)