# Lesson 3-7:
# Course Conclusions

# Lambda Expressions

- Lambda expressions give us a simple way to define behaviour
  - Can be assigned to a variable or passed as a parameter
- Can be used wherever the type is a functional interface
  - One that has only one abstract method
  - The lambda expression provides an implementation of the abstract method

# Stream API

- Pipeline of operations to process collections of data
  - Multiple sources, not just from the Collections API
  - Can be processed sequentially or in parallel
- Sources, intermediate and terminal operations
- Behaviour of intermediate and terminal operations often defined using Lambda expressions
- Terminal operations often return an `Optional`
- We can now use a functional style of programming in Java

# Lambdas And Streams: Think Differently

- Need to think functional rather than imperative
  - Try to stop thinking in loops and using mutable state
- Think of how to approach problems using recursion
  - Rather than an explicit loop
  - Avoid `forEach` (except for special cases)
- Infinite streams don't need to be infinite
- Remember, parallel streams always involve more work
  - Sometimes they complete the work quicker

# Thank You!

## This Is The End Of The Course

ORACLE®