Week 11 Research

Q) What are streams?

A) Streams in Java are a powerful tool for manipulating data. It is often described as a pipeline or a conveyor belt. This is because data is put into the stream and modified as many times as you want, each “step” of the pipeline does its thing to it, and passes the new data on to the next stage. It repeats this until the stream has executed all of the operations and methods that were set up, returning the result.

The great thing about streams is that you see the result without changing the data. That’s why above, I described it as manipulating, because you can dress it up and make temporary changes, but it is just “clothing” and it returns to its original state. Another benefit of using streams is that the functions and methods are “lazily” executed, meaning they only get processed as the result that would the processing is expected. They are very efficient and make great use of newer multi-core processors.

Q) What are Lambda Expressions?

A) Lambdas are code blocks which take a set of parameters and return a value, similar to some methods, but have different rules and syntax. Some developers comment that lambdas are just a different way to scramble an egg so to say, but they have certain features that methods can’t duplicate exactly.

One such use, they can be placed inside variables. They can only do this for interfaces with a single method, but can achieve the result of several functions in succession. Lambdas also allow us to create methods that are not class based but can still be passed.

A great feature of Lambdas is Method References. Using existing methods, a quick bit of code can create a short, clear, functional use of programming. Instantiated in this way, method references have the power and reusability of methods, combined with the fluid and readable Lambda syntax. You can even reference constructors this way.

<https://www.geeksforgeeks.org/stream-in-java/>

<https://www.simplilearn.com/tutorials/java-tutorial/streams-in-java>

<https://stackify.com/streams-guide-java-8/>

<https://www.w3schools.com/java/java_lambda.asp>

<https://medium.com/javarevisited/java-lambda-expressions-6696793a6c38>

<https://www.tutorialspoint.com/java8/java8_lambda_expressions.htm>