Lambda Calculus

Lambda Calculus comes under formal system in mathematical logic.

Formal System in programming language has 2 aspects:

Syntax – what the language looks like.

Semantics – what is the meaning of words, signs and symbols.

Lambda Calculus is used for expressing based on abstraction and application using variable binding and substitution.

To make semantics simple λ-calculus incorporates two simplifications.

First is that it treats the functional anonymously without explicit names. For example

cubeSum(x, y, z) = (x \* x \* x) + (y \* y \* y) + (z \* z \* z)

can now be rewritten in anonymous form as

(x, y, z) → (x \* x \* x) + (y \* y \* y) + (z \* z \* z)

Second simplification is that we can chain functions inside functions.

x → (y →(some formulae))(parameter1)) (parameter2)

This is basics of Lambda Calculus. This post was intended to be just the basic introduction for Lambda.

In next post we will learn about Lambda Operator in Java, Functional Interface, How to use Lambda and when to use Lambda?