CASE PROGRAM – CALLER RESTRICTIONS

package myentitites

import case.lang.System

namespace EntitiesNameSpace {

String->Object->Main

#public class Program

[public Program(String [] args)

[EntityPool Pool = EntityPool.getEntityPool]

assert(Pool) //asserts that Pool exists and has a value

Stream (n) String

]

**//Pass by Value**

//Passes a copy of a primitive or an ENTIRE object – including deep copy.

functionName(int parameter)

functionName(deep parameter)

**//Pass by Need**

//This uses delay and force actions like in Haskell. In this case, a promise is passed //and it makes due on that promise when the variable is needed.

functionName(need parameter)

**//Pass by Reference**

//This is called when there is a large object and it doesn’t make sense to pass an //entire copy of it.

functionName(&parameter)

**//Pass by Reference by Value**

Like Java, variables can be passed into a function altered and then upon closing of that function, the variables data is kept and passed back to the calling function.

functionName(parameter)

#end class

}