CASE PROGRAM – USING INCLUSIONS

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

Int MyInt = EntityPool.getStreamMemory() //retrieve mem from pool

Int GetInt = EntityPool.get(“MyInt”) //pointer to MyInt using pool get

//get pointer to CurrentLocationInList from the pool

Int ListStatus = n.get(“CurrentLocationInList”)

//Often times you have to initialize content within for a for loop, for //example, but you can’t because it is dependent on the loop.

//A java example

<String> output = **null**;

**for** (<XMLPatternTag> tags : source.XMLPatternType)

{

output = output + tags.name;

}

In this case we want to iniitalize output to some value that’s not **null**, but is the first element of XMLPatternType. Otherwise, the string is initiailized to null and then the rest of the loop content is carried over. This is a common problem . The **inclusion** keyword is meant to solve this. It is an essentially an initializer that executes code that only executes the first iteration of the loop.

The **inclusion** code is the only code that executes the very first iteration and it only executes once.

The solution is:

**for** (<XMLPatternTag> tags : source.XMLPatternType)

{

inclusion output = output + tags.name;

output = output + tags.name;

rename output => stringout//dynamically rename output to stringout

Print stringout

Print output

//Able to rename one class to something else dynamically

//rename <BobsClass> => <TedsClass>

}

]

#end class

}