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”)

]

[public void FunWIthArrays()

**Ellipses**

**<array[size] variablename = braced list**

<array[5]> myArray = {1, 2, 3, 4, 5}

>>myArray[1..3]

[1, 2, 3]

<array[5]> myArray2 = { a, b, c, d, 1, 3, 2 }

>>myArray2[1...5]

[a,b,c,d,1]

**storing code using an array**

**Storing code in an Array**

>>[ for int i=0;i<size;i++;print i]=myLoopArray

>>[myLoopArray] //calls myLoopArray

>>1

>>2

>>3

>>4

>>5

>>6]

[public void multidimensionalarrays()

//array

<int []> myArray = new <[]>// Always use the data type and //brackets

//matrix

int <[][]> myArray = new <[][]>;

Multi-dimensional arrays allocate their own memory by need.

Oblong Array: <int [5][]> myArray = new <int[5][]>;

//Two ways to initialize this.

//First:

for (int myCounter:myArray)

{ myCounter[] = new int[] }

//Second:

myArray = new <int[5][5]>

//Accessing Oblong Array:

myArray[x][y]

//Accessing 2-dimensional Array:

myArray[1][3]

//Accessing 3—dimensional Array:

myArray[0][1][8]

//You can use as many dimensions as you like.

]

#end class

}