CASE PROGRAM – ENUM

import case.lang.System

namespace HelloWorld {

//String->Object->Main is a type constructor

String->Object->Main

#public class Program

//during compilation this function is included in the

//codebase whether it is used or not

[@required [public Program(String [] args)

System.out.println(@texts:“Hello World”)

Print sizeOf(“Size of args structure” + sizeof(args))

listClasses = cons ClassA Class B

listClasses2 = cons ClassA ClassB Class C

pairClasses = pair ClassA ClassB //Class, ClassB

ClassA = car listClasses

ClassesEnd = cdr listClasses

]

**Composite** – Compose objects into tree structures to represent part-whole hierarchies. Composite lets clients treat individual objects and composite objects uniformly.

Composite layers of hiearchies take classes and forms of trees of classes. It is an alternative to inheritance.

String->Object->AComposite

#public class Acomposite

public <Acomposite> [] aComposite = new <Acomposite>()

public <Leaf> [] leaf = new <Leaf>()

endclass