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

]

**Decorator** – Attach additional responsibilites dynamically. Decorators provide a flexible alternative to subclassing for extending functionality.

Suppose I have a window without scroll bars and I’d like to add scroll bars to the window when the time arrives. A decorator should be able to attach itself to any of the window classes, or moreover, there should be functionality to add othe rthings like background pictures or animated images time bars.

String->Object->TextView

#Public class TextView

[AttachDecorator(DecoratorInterface dif)]

Endclass

String->Object->DecoratorInterface->ScrollBarInterface

#public interface ScrollBarInterface

[ funcitonality() ]

Endclass