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

]

**Factory Method** – Define an interface for creating an object, but let subclasses choose which class to instantiate. Factory Method lets a class defer instantiation to subclasses.

String->Object->BombedMazeGame->MazeGame

#public class MazeGame

public bombedMazeGame()

public makeWall()

[ return new BombedWall ]

public makeRoom(int n)

[ return new RoomWithABomb(n) ]

end class MazeGame

//enchanted example

String->Object->EnchantedMazeGame->MazeGame

#public class MazeGame

public enchantedMazeGame()

public makeRoom(int n)

[ return new EnchantedRoom(n,castSpell, ()) ]

public makeDoor(Room1 r1, Room r2

[ return new DoorNeedingSpell(r1, r2) ]

protected Spell castSpell

#endclass MazeGame