CASE PROGRAM – BRACES

import case.lang.System

namespace HelloWorld {

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

String->Object->Main

#public class Program

[public Program(String [] args)

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

]

**//**Classes do not use braces but use # notation to start and end a class. //Braces for a class are messy and many a program has scrolled up and //down trying to ifgure out their class braces.

//Braces are somewhat flexible. For functions, the brace is always a square //bracket, i.e,

[MyFunction()

Integer int + 1

]

//Unimplemented functions will compile with a warning unless you try to //call on them. So the following is legal:

[MyFunction()]

//or

[public void MyFunction()]

//We want functions to stand out, they are the workhorse of the application //and shouldn’t be equivalent to control structures (for, while, until, etc.)

//Braces for things like mentioned above are for control structures priarily //like for, while, until, do-while, foreach, etc. There are several way to use the //braces.

//For a single statement:

for(Integer A;A<100;A++)

{ Print A }

//For multiple statements we can write that as one line as well, delimiting //with a comma.

For (IntegerA;A<100;A++)  
 { Print A, MyFunction(), stream (n) Float, MyFloat (n) Float}

//We can merge the two as well

For (IntegerA;A<100;A++)  
{{ Print A, MyFunction(), stream (n) Float, MyFloat (n) Float} MyFunction(MyFloat), atom TestVariable}

//Or we can do the standard notation

For (Integer A;A<100;A++)

{

Print A

MyFunction()

Stream (n) Float

MyFloat (n) Float

` MyFunction(MyFloat)

atom TestVariable

}

//The top version is more compact, and in the spirit of the language more //appropriate as we are aiming for compactnesss. The very bottom example is a //little bit more legible.

#end class

}