1 Definitions

scope  The *scope* of a named binding is the textual region of the program in which the binding is visible.

referencing environment  The *referencing environment* at any program point is the set of bindings visible at that point.

namespace  A referencing environment is partitioned into *namespaces*. Each lookup operation will look up identifiers in only one namespace. As a result, two bindings with the same name but different namespaces do not conflict.

2 Example Program

class Foo(var z : Int)
{
    var x : Int = 0;

    def m(z : Any) : String =
        z match {
            case zz : Main => "Caught you!"
            case zz : Any => "Nevermind"
        };
}

class Main() extends Foo(3)
{
    var y : Boolean = false;

    {
        new IO().out(new Foo().m(this).concat("\n"))
    }
}

3 Questions

(a) What is the *referencing environment* at the point “Nevermind.”

(b) What is the *scope* of the formal parameter z?

(c) What *namespaces* does the Cool referencing environment have?