Assignment 1 Review: Comparison Operators

Current Semantics:

1 < 3 returns 0

1 > 3 returns Null

Distinguished True Type:

1 < 3 returns True

1 > 3 returns Null

Interpret Numerical Values as Booleans:

1 < 3 returns 1

1 > 3 returns 0

Meaningful Value for True:

1 < 3 returns 1

2 < 5 returns 2

1 > 3 returns Null

Assignment 1 Review: Constructors

```
defn cplx (r, i) :
   object :
      var real = r
      var imag = i
      …

var c = cplx(10, 10)
```

```
public Cplx (int r, int i) {
   this.real = r
   this.imag = i
}

Cplx c = new Cplx(10, 10)
```

Assignment 1 Review: Operator Overloading

```
var a = cplx(10, 10)
var b = cplx(10, -5)
var c = cplx(5, -5)

(a + b) * c

a.add(b).mul(c)
```

Assignment 1 Review: Inheritance

```
defn make-barkable (x) :
   object(x) :
      method woof () :
         println(“Woof!”)

val c = cplx(10, 10)
val c2 = mark-barkable(c)
c2.woof()
```

Assignment 1 Review: Default Values

```
object :
   var a = exp1
   var b = exp2

var x = exp3
…
```

Assignment 1 Review: Closures

```
defn f (x) :
   fn (i) :
      x + i

var g = f(10)
g(11)
```

```
defn f (x) :
   object :
      var x = x
      method call (i) :
         this.x + i

var g = f(10)
g.call(11)
```

Assignment 1 Review: Closures

```
var g = null
var h = null

defn f (x) :
   g = fn (i) :
          x + i
   h = fn (i) :
          x - i

f(10)
g(1)
h(1)
```

```
var g = null
var h = null

defn f (x) :
   g = object :
      var x = x
      method call (i) :
         this.x + i
   h = object :
      var x = x
      method call (i) :
         this.x - i

f(10)
g.call(1)
h.call(1)
```

Assignment 1 Review: Closures

```
var g = null
var h = null

defn f (x) :
   g = fn (i) :
         x = x + i
         x
   h = fn (i) :
         x = x - i
         x

f(10)
g(1)
h(1)
```

```
var g = null
var h = null

defn f (x) :
   g = object :
      var x = x
      method call (i) :
         this.x = this.x + i
         this.x
   h = object :
      var x = x
      method call (i) :
         this.x = this.x - i
         this.x

f(10)
g.call(1)
h.call(1)
```

Assignment 2: Machine State

The Global Variable Map :

x = 40

y = 10

z = null

…

The Current Local Frame :

Parent Frame = …

Return Address = …

Slot 0 = 10

Slot 1 = 20

Slot 2 = null

…

The Operand Stack :

20, 30, 40, null, 10, …

Assignment 2: "Quickening"

1. Resolve global variables to an integer index.
2. Lay out all instructions in a linear array. The "address" of an instruction is the index of its location in the array.
3. Allocate local frames in an array. The pointer to a frame (also called the frame pointer) is just the index at which the frame starts.
4. Resolve labels for control flow instructions ahead of time.