CS 164, Fall 2020 **CS 164: Homework #6** **P. N. Hilfinger**

Due: Friday, 13 November 2020

1. When executing the following Python program:

```
def f0(x1):
    def f1(h2):
        return h2(7)
    def g1(x2):
        def f2(n):
            # Stop Here
            return x2 + n
        return f1(f2)
    return g1(15)
print(f0(3))
```

show what the frames in the execution stack will contain when execution reaches the point marked "`# Stop Here`". Use the template for this problem in `hw6.txt`. Place all frames on the stack that can validly reside there.

2. Because Chocopy requires that all names have explicit static types and does not have a way to denote function types, it is not possible to return or store function values in Chocopy. In Python, however, it is. Consider the following program:

```
def f0(x1):
    def f1(h2):
        return h2(7)
    def g1(x2):
        def f2(n):
            # Stop Here
            return x2 + n
        return f1(f2)
    return g1
print(f0(3)(11))
```

Show the execution stack and heap contents when execution reaches the point marked "`# Stop Here`."

3. Fill in the skeleton file `oop.py` to create a utility for use by a very simple runtime system for an object-oriented language with single inheritance and no interfaces. The utility defines a class `Linker`, which is intended to be used to take a set of class definitions and create appropriate virtual tables for instances of these classes. Replace just the `# FIXME` comments so that the two (already defined) functions `new_instance` and `call_method` work as intended. All the method bodies are functions taking an instance as a "self" parameter.

The idea is for you to deduce from the implementations of these two functions what the format and contents of the virtual tables are supposed to be.