

Homework 03 - Project Revision

Andrew Carter and Beryl Egerter

February 10, 2013

- 1 Scheduled Interviews**
- 2 Description of Topic**
- 3 Empirical Goals**
- 4 Response to Participant Questions**
- 5 Non-Linear Interview Questions**
- 6 Interview Sequence and Questions**
- 7 Handouts**

Attached below.

Question 1.1

```
def func2(list, num):
    return func1(list, num, func4)

def func4(a, b):
    return a * b

def func1(list, num, f):
    acc = 0
    for i in list:
        acc += f(i, num)
    return acc

def main():
    print(func3([1,2,3,4]))

def func3(list)
    return func2(list, 4)

main()
```

Question 1.2

```
def function50(i, L):  
    return L[i+2]  
  
def function37(L):  
    return L[-1]+L  
  
def function52(i):  
    return function4() * i  
  
def function1(j, k):  
    return (j + k) * function52(1)  
  
def function4():  
    return 3  
  
def function188(L):  
    return function37(L)+function50(2, L)  
  
def function0():  
    return function188([1,2,3,4,5,6,7,8,9])[function1(0,1)]  
  
function0()
```

Question 2.1, File 1

```
#!/usr/bin/ruby

load "ourdate.rb"

d = OurDate.new(2011,1,4)
print "#{d.what_day}"
print "We started writing this file today.\n"
d.forward_time(365)
print "We are almost done now.\n"
print "#{d.what_day}"
```

Question 2.1, File 2

```
#!/usr/bin/env ruby

$months31 = [1,3,5,7,8,10,12]
$months30 = [4,6,9,11]

class OurDate
  attr_accessor :year
  attr_accessor :month
  attr_accessor :day

  def initialize(year, month, day)
    @year = year
    @month = month
    @day = day
  end

  def is_equal?( d )
    puts @year == d.year and
          @month == d.month and
          @day == d.day
  end

  def is_leap_year?
    if @year % 400 == 0
      puts 1
    elsif @year % 100 == 0
      puts nil
    elsif @year % 4 == 0
```

```

        puts 1
    else
        puts nil
    end
end

def check_month
    if @month == 13
        @month = 1
        @year = @year + 1
    elsif @month == 0
        @month = 12
        @year = @year - 1
    end
end

def tomorrow
    @day = @day + 1
    if @day > 31
        for i in $months31
            if @month == i
                @day = 1
                @month = @month + 1
                check_month
            end
        end
    elsif @day > 30
        for i in $months30
            if @month == i
                @day = 1
                @month = @month + 1
            end
        end
    end
end

```

```

        check_month
    end
end
elseif @day > 28 and @month == 2
    @day = 1
    @month = @month + 1
    check_month
end
end

def yesterday
    @day = @day - 1
    if @day == 0
        @month = @month - 1
        check_month
        for i in $months31
            if @month == i
                @day = 31
            end
        end
        for i in $months30
            if @month == i
                @day = 30
            end
        end
        if @month == 2
            @day = 28
        end
    end
end
end

```

```
def forward_time(n)
  for i in 0..n
    tomorrow
  end
end

def reverse_time(n)
  for i in 0..n
    yesterday
  end
end

def what_day
  puts "Today is #{month}/#{day}, #{year}!"
end
end
```