CSc600 – Homework: Ruby

Dr. Jozo Dujmović

Hin lok chan

12/12/16

Q1

#if

x = "if"

if x == "if"

puts "this is example of if"

end

#if else

y = "1"

if y == "1"

puts "example of if\_else (true)"

else

puts "example of if\_else (false)"

end

#if elsif else

z = "1"

if z > "1"

puts "example of if\_elsif\_else (if)"

elsif z < "1"

puts "example of if\_elsif\_else (if\_elsif)"

else

puts "example of if\_elsif\_else (else)"

end

#if modifier

puts "example of if modifier" if z == "1"

#unless

a = 3

b = 5

unless x < y

puts "example of unless"

end

#unless else

unless x < y

puts "example of unless else (true)"

else

puts "example of unless else (false)"

end

#unless modifier

puts "example of unless modifier" unless z != "1"

#case

c = "1"

num = case c

when "1" then "one"

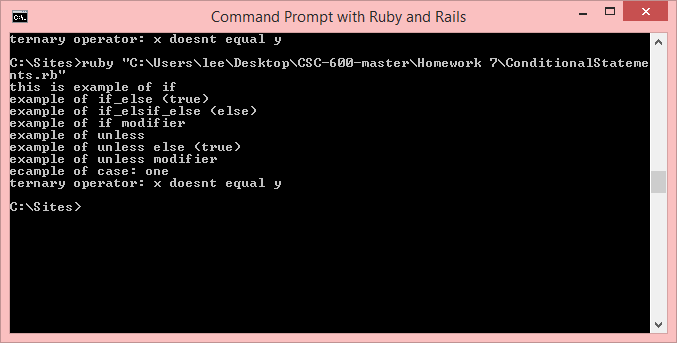
when "2" then "two"

when "3" then "three"

else "error"

end

puts "ecample of case: #{num}"



Q2

#loop do

print "example of loop do: "

i = 0

loop do

if i > 10

break

end

print "#{i = i + 2} "

end

puts

#until do

print "example of until do: "

j = 0

until j == 10 do

print "#{j} "

j = j + 2

end

puts

#while do

print "example of while do: "

k = 0

while k < 10 do

print "#{k} "

k = k + 2

end

puts

#while modifier

print "example of while modifier: "

l = 0

print "#{l = l + 2} " while l < 10

puts

#unto modifier

print "example of unto modifier: "

array = [0,2,4,6,8]

print array.pop.to\_s + " " until array.empty?

puts

#for in

print "example of for in: "

hash = {:Zero => "0", :Two => "2", :Four => "4"}

for key,var in hash

print "#{key} => #{var}, "

end

puts

#upto

print "example of upto: "

1.upto(8){|m| print "#{m} "}

puts

#downto

print "example of downto: "

8.downto(1){|n| print "#{n} "}

puts

#times

print "example of times: "

4.times{print "tester "}

puts

#each

print "example of each: "

array = [0,2,4,6,8]

array.each{|o| print "#{o} "}

puts

#map

print "example of map: "

p = 0

(1..8).map{|p| print "#{(p\*p)} "}

puts

#step

print "example of step: "

0.step(8,2) do |q| print "#{q} " end

puts

#collect

print "example of collect: "

squares = [0,2,4,6,8].collect{|r| r}

print squares

puts

#select

print "example of select: "

evens = (0..8).select{|s| s%2 == 0}

print evens

puts

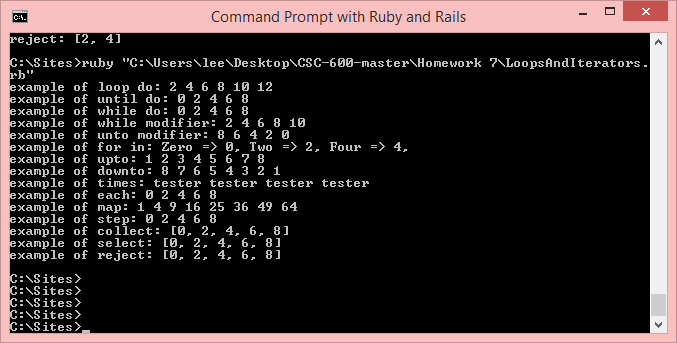
#reject

print "example of reject: "

evens = (0..8).reject{|t| t%2 == 1}

print evens

puts



Q3

def mean\_sigma(v)

sum = 0.0

v.each {|i| sum = sum + i}

mean = sum/v.length

sum = 0.0

v.each {|j| sum = sum + (j - mean)\*\*2 }

std = Math.sqrt(sum/(v.length-1))

return mean,std

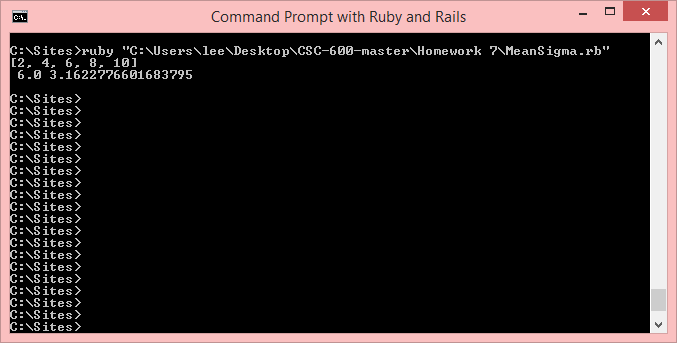
end

k = [2,4,6,8,10]

mean,std = mean\_sigma(k)

print "#{k}\n"

puts " #{mean} #{std}"



Q4

def sort(v)

temp = v.clone

sort =[]

x=0

while x < v.length do

del = 0

min = temp[0]

y=0

while y < temp.length do

if temp[y] <= min

min = temp[y]

del = y

end

y = y+ 1

end

sort.push(min)

temp.delete\_at(del)

x = x + 1

end

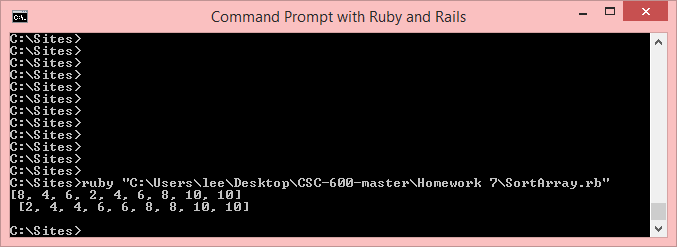
return sort

end

array = [8,4,6,2,4,6,8,10,10]

print "#{array}\n"

print " #{sort(array)}\n"



Q5

class Triangle

attr\_accessor :x, :y, :z

def initialize(x,y,z)

@x,@y,@z = x,y,z

end

def test

if @x != "0" && @y != "0" && @z != "0"

if @x == @y && @x == @z

return "equilateral"

elsif (@x == @y && @x != @z) || (@y == @z && @x != @z) || (@x == @z && @y != @z)

return "isosceles"

elsif @x\*\*2 == @y\*\*2 + @z\*\*2 || @y\*\*2 == @x\*\*2 + @z\*\*2 || @z\*\*2 == @y\*\*2 + @x\*\*2

return "right"

elsif @x != @y && @x != @z && @z != @y

return "scalene"

end

else

return "not a triangle"

end

end

def per

return @x+@y+@z if self.test != " not a triangle"

end

def area

return Math.sqrt((self.per/2)\*( (self.per/2)-@x)\*( (self.per/2)-@y)\*( (self.per/2)-@z)) if self.test != " not a triangle"

end

end

a = Triangle.new(1,1,1)

print " #{a.x} #{a.y} #{a.z} \n"

puts "#{a.test} perimiter: #{a.per}, area: #{a.area}"

b = Triangle.new(1,1,2)

print " #{b.x} #{b.y} #{b.z} \n"

puts "#{b.test} perimiter: #{b.per}, area: #{b.area}"

c = Triangle.new(1,2,3)

print " #{c.x} #{c.y} #{c.z} \n"

puts "#{c.test} perimiter: #{c.per}, area: #{c.area}"

d = Triangle.new(5,3,4)

print " #{d.x} #{d.y} #{d.z} \n"

puts "#{d.test} perimiter: #{d.per}, area: #{d.area}"

