Handout:

Contributors: Cole Albers (Created), David Sinchok, Andrew Fox, Adam Prater, and Sam Yinger.

Learning Objectives:

1) Understand what RSpec is and why it is useful.

2) Understand how to install RSpec and create a basic Hello World! example.

3) Understand the basic syntax of RSpec.

4) Understand how to write RSpec specs.

Hello World! Example:

#Beginning of Ruby Code.

class HelloWorld

def hello\_world

‘Hello World!’

end

end

#End of Ruby Code.

#Beginning of RSpec Code.

describe HelloWorld do

context “when testing the HelloWorld class” do

it “should return ‘Hello World!’ when the hello\_world method is called.” Do

hello\_world\_object = HelloWorld.new

output = hello\_world\_object.hello\_world

expect(output).to eq ‘Hello World!’

end

end

end

#End of RSpec Code.

Using metadata to limit variable scope:

#Beginning of Ruby Code.

describe Array.new, "Example of using metadata to scope", :var1 => true do

context "with a context for further layers of scoping", :var2 => 45 do

it "will have access to var1, var2, and its own variable, var3", :var3 => "Test" do |example|

#Accessing the metadata hash with valid keys

puts example.metadata[:var1]

puts example.metadata[:var2]

puts example.metadata[:var3]

#Using the subject

subject[0] = "Hello"

subject[1] = "World"

subject.clear

end

it "will not be able to use var3, but can use var1 and var2." do |example|

subject { [1,2,3] }

#Accessing the metadata hash with valid keys

puts example.metadata[:var1]

puts example.metadata[:var2]

#Will only print a blank line, because the key var3 is not associated with any value here.

puts example.metadata[:var3]

end

end

end

#End of Ruby code.