# Software Engineer (Ruby) Examination

The purpose of this examination is to assess the capabilities and training of candidates for the **Software Engineer** position. It is broken down into several parts. There are three types of questions:

1. Free-form questions – these questions require that you type in the answers to the questions in paragraph form. Answer these as best you can.
2. Code questions – these questions require you to write code. Answers to these questions will not be judged only on correctness, but on code clarity, comments, style, and elegance. They will always be instructions on how to label the code that you submit back to us (i.e., submit as **infinite\_sum.rb** ).

Draw upon any resources necessary to answer these questions. State any assumptions you've made if the question is not clear. When you submit your answer, please submit one (1) Word document for your free form answers, and one zip file containing all of the source code answers. Please respond with your submission within 24 hours.

Finally, the questions are based on pure Ruby and should not assume there are additional libraries being used. Ex: Rails, ActiveRecord, ActiveSupport, etc.

## ***Problem Solving***

1. You receive an exception notice from a piece of code running in production. It is an **NoMethodError** exception. Looking at the stack trace, you see an error happens on line 64. Here is the code:

59: def pack\_basket\_with\_fruit()

60: b = get\_basket

61: f = get\_fruit\_from\_basket(1)

62: seeds = f.seed\_count

63:

64: b.add(f, f.seed\_count)

65:

66: b.pack\_basket

67: end

Based on this code – what is the cause of the error?

1. Consider the following code:

1: p = nil;

2: name = ["Andrew Ryan", "Jonathan White", "Al Owens"]

3: for i in 0..3

4: if i < 1

5: puts "First Name" + name[i]

6: else

7: puts "First Name length" + p.length

8: end

9: end

This code has two, big, BIG bugs that will cause exceptions. What are those exceptions, and what lines of code will cause them? What is wrong with the general approach used above to iterate through a collection of objects in Ruby (hint: a different approach would avoid one of the two bugs)?

## ***XML***

1. Consider the following XML:

<?xml version="1.0" encoding="UTF-8"?>

<Books>

<Book Name="Moby Dick" Genre="Classic">

<Price>$3.95</Price>

</Book>

<Book Name="Nightmare on Elm Street" Genre="Horror">

<Price>$13.95</Price>

</Book>

<Book Name="Friday the 13th" Genre="Horror">

<Price>$2.95</Price>

</Book>

<Book Name="Gone with the Wind" Genre="Drama">

<Price>$33.95</Price>

</Book>

</Books>

Write the following xpath expressions:

1. Find all **Horror** books
2. Find all books over $10
3. Find all books that are NOT Drama

## ***Algorithms***

1. Given a Zip code (Postal Code) database containing 43,000 entries and the code below:

1: def get\_state\_by\_zipcode(zipcode)

2: raise ArgumentError.new('zipcode is nil') if zipcode.nil?

3: # Get all locations from text file

4: locations = get\_zipcode\_locations()

5:

6: location = locations.find { |l| l.zipcode == zipcode }

7: location && location.state || nil

8: end

1. How might you improve the performance?
2. Compare the performance gains, what makes your solution better?

## ***Ruby***

1. What is the difference between a process, and a thread?
2. Explain why the code below will cause problems when run across multiple threads, and rewrite it so that it is thread safe: **(This is a code question, name the file threadsafe.rb)**

10: MAX\_MESSAGES = 100

11: messageCounter = 0

21: def process\_message(msg)

22: if( msg.valid? && message\_counter < MAX\_MESSAGES )

23: msg.file

24: message\_counter += 1

25: end

26: end

1. Using the code above (not the threadsafe.rb code), describe the test cases you would implement to ensure the method is properly tests.
2. Refactor the method below to be more readable by your peers: **(This is a code question, name the file censorship\_refactor.rb)**

1: def unnecessary\_censorship\_sentence(input)  
2: ctr = 0  
3: input.each do |word|  
4: word = word.downcase  
5: if word == "happy" || word == "orange" || word == "pear"  
6: i=0  
7: word.length.times do  
8: input[ctr][i,1] = "\*"  
9: i+=1  
10: end  
11: end  
12: ctr += 1  
13: end  
14:  
15: puts input.join(" ").to\_s  
16: end

1. Consider the following code:

1: def parse\_fields(note\_attributes)

2: note\_attributes.map do |attrib|

3: case attrib.name

4: when 'requested\_ship\_date'

5: {

6: requested\_ship\_date: try\_format\_datetime(attrib.value, '%Y-%m-%dT%H:%M:%SZ')

7: }

8: end

9: end.compact.reduce({}, :merge)

10: end

1. Explain the intention of this code?
2. Explain how you might refactor this code?
3. What will the following code output?

s = "Hello World"; x = nil

puts "#{(s || 'No string')} is a test"

s = x

puts "#{(s || 'No string')} second try"

1. Write a function that accepts an array of numbers and returns their product. (**Name your file product.rb)**

## ***SQL***

1. Consider the following Tables:

Customers

ID NAME ADDRESS PHONE\_NUMBER EMAIL

Orders

ID CUSTOMER\_ID ORDER\_AMOUNT ORDER\_ADDRESS

1. Write the SQL Query to pull all customers
2. Write a SQL Query to pull all customers that have orders (no duplicates)
3. Write a SQL Query to pull all customers that do NOT have orders
4. If you had to create an index on these tables, which fields would you most likely index and why?
5. Write a query that lists each customer name, email, and the number of order they have
6. Write query that pulls all customers with between 2 and 5 orders.

## ***Design Patterns***

1. Describe the **Strategy** pattern and when you use it, giving a concrete example.
2. Describe the **Unit Of Work** pattern, and what problems it solves.