**Quiz 5**

1. Write a Ruby program that Reads from sample.txt (contains plain text) and creates a new config file called text.cfg

require 'yaml'

configuration = { 'color' => 'blue',

                 'font' => 'new romain',

                 'font-size'  => 7 }

open('text.cfg', 'w') { |f| YAML.dump(configuration, f) }

open('text.cfg') { |f| puts f.read }

open('text.cfg') { |f| YAML.load(f) }

**Sample output:**

---color: blue                                                        font: new romain                                                   font-size: 7

2. Write a sample ruby program that uses XML and JSON parsing

class Parser

 def parse

   puts 'The Parser class received the parse method'

 end

end

class XmlParser

 def initialize(parser)

   @parser = parser

 end

 def parse

   @parser.parse

   puts 'An instance of the XmlParser class received the parse message'

 end

end

class JsonParser

 def initialize(parser)

   @parser = parser

 end

 def parse

   puts 'An instance of the JsonParser class received the parse message'

   @parser.parse

 end

end

puts 'Using the XmlParser'

parser = Parser.new

XmlParser.new(parser).parse

puts 'Using the JsonParser'

JsonParser.new(parser).parse

puts 'Using both Parsers!'

JsonParser.new(XmlParser.new(parser)).parse

**Sample Output:**

Using the XmlParser

The Parser class received the parse method

An instance of the XmlParser class received the parse message

Using the JsonParser

An instance of the JsonParser class received the parse message

The Parser class received the parse method

Using both Parsers!

An instance of the JsonParser class received the parse message

The Parser class received the parse method

An instance of the XmlParser class received the parse message

=> nil

3. Write a sample program that converts a string or array to YAML ?

require 'yaml'

HASH\_OF\_HASHES = {

 "hostname1.test.com"=> {"public"=>"51", "private"=>"10"},

 "hostname2.test.com"=> {"public"=>"192", "private"=>"12"}

}

puts HASH\_OF\_HASHES.to\_yaml

puts

ARRAY\_OF\_HASHES = [

 {"hostname1.test.com"=> {"public"=>"51", "private"=>"10"}},

 {"hostname2.test.com"=> {"public"=>"192", "private"=>"12"}}

]

puts ARRAY\_OF\_HASHES.to\_yaml

**Sample Output:**

---

hostname1.test.com:

 public: '51'

 private: '10'

hostname2.test.com:

 public: '192'

 private: '12'

---

- hostname1.test.com:

   public: '51'

   private: '10'

- hostname2.test.com:

   public: '192'

   private: '12'

=> nil

4. Write a Program that converts string to YAML and shows difference between different ways of YAML conversion.

require 'yaml'

str = %{This string's a little complicated, but it "does the job" (man, I hate scare quotes)}

puts str.inspect

puts str.to\_yaml

puts str.inspect.to\_yaml

**Sample output:**

sh-4.3$ ruby main.rb

"This string's a little complicated, but it \"does the job\" (man, I hate scare quotes)"

--- This string's a little complicated, but it "does the job" (man, I hate scare quotes)

...

--- '"This string''s a little complicated, but it \"does the job\" (man, I hate scare

  quotes)"'

5 .Write a sample ruby program for YAML File  and try to modify YAML file

require 'yaml'

foo = {"file.yaml" => ["extra","intra","lateral"]}

bar = foo.to\_yaml

puts bar

baz = YAML.load(bar)

baz

foo.each do |k, v|

 File.write(k, v.to\_yaml)

end

foo = YAML.load\_file('file.yaml')

**Sample Output:**

Before modifying

---

file.yaml:

- extra

- intra

- lateral

=> {"file.yaml"=>["extra", "intra", "lateral"]}

After modifying:

---

file.yaml:

- extra

- intra

- lateral

=> ["extra", "intra", "lateral"]

6. Write a Sample ruby program for Parsing YAML

tree = YAML::parse( File.open( "File" ) )

puts tree.type\_id

title = tree.select( "/title" )[0]

puts title.value

obj\_tree = tree.transform

puts obj\_tree['title']

Sample Output:

map

YAML.rb

YAML.rb

7. Write a program using YAML which checks for user id and password

require 'yaml'

yml = YAML.load\_file 'userdetails.yml'

yml.each\_key { |key|

 username = yml[key]['user']

 password = yml[key]['pass']

 puts "#{username} => #{password}"

}

8. Write a sample program that connects to database and create a table in database

require 'dbi'

dbh = DBI.connect('DBI:OCI8:ORCL', 'hr', 'hr')

sqlInsert = "INSERT INTO states (id, name, capital)

                        VALUES (?, ?, ?)"

dbh.do(sqlInsert, "AL", "Alabama", "Birmingham")

dbh.do(sqlInsert, "AZ", "Arizona", "Phoenix")

dbh.do(sqlInsert, "CO", "Colorado", "Denver")

dbh.commit

9. Write a program that creates a connection to an HTTP server. Use net/http

require 'net/http'

uri = URI('http://example.com/some\_path?query=string')

Net::HTTP.start(uri.host, uri.port) do |http|

 request = Net::HTTP::Get.new uri.request\_uri

 response = http.request request # Net::HTTPResponse object

end

**Explanation:** Net::HTTP::start immediately creates a connection to an HTTP server which is kept open for the duration of the block. The connection will remain open for multiple requests in the block if the server indicates it supports persistent connections.

9. Write another sample program that uses NET::HTTPS

require "net/http"

require "uri"

uri = URI.parse("http://google.com/")

# Shortcut

response = Net::HTTP.get\_response(uri)

# Will print response.body

Net::HTTP.get\_print(uri)

# Full

http = Net::HTTP.new(uri.host, uri.port)

response = http.request(Net::HTTP::Get.new(uri.request\_uri))