**31. Write if/else statement in only one line**

**Program:**

puts 1 < 2 ? "one is less than two!" : "one is not less than two."

puts 5 >= 5 ? "true" : "not true"

**32. What is the sum of all the numbers between 0 and 100 that are divisible by both 3 and 5?**

**Program:**

(0..100).to\_a.inject(0) do |memo, number|

memo += number if number % 3 == 0 && number % 5 == 0

memo

end

# OR

(0..100).to\_a.select do |number|

number % 3 == 0 && number % 5 == 0

end.inject do |memo, number|

memo += number

memo

end

**33. Write a program to convert user input to upper, lower and capital letters**

**Program:**

print "write something!: "

user\_input = gets.chomp

puts user\_input.downcase!

puts user\_input.upcase!

puts user\_input.capitalize!

**34. Write a program to print numbers from 1 to n using until or while loop**

**Program:**

i = 1

while i <= 10 do

puts i

i += 1

end

k = 1

until k == 10 do

puts k

k += 1

end

**31. Write if/else statement in only one line**

**Program:**

puts 1 < 2 ? "one is less than two!" : "one is not less than two."

puts 5 >= 5 ? "true" : "not true"

**32. What is the sum of all the numbers between 0 and 100 that are divisible by both 3 and 5?**

**Program:**

(0..100).to\_a.inject(0) do |memo, number|

memo += number if number % 3 == 0 && number % 5 == 0

memo

end

# OR

(0..100).to\_a.select do |number|

number % 3 == 0 && number % 5 == 0

end.inject do |memo, number|

memo += number

memo

end

**33. Write a program to convert user input to upper, lower and capital letters**

**Program:**

print "write something!: "

user\_input = gets.chomp

puts user\_input.downcase!

puts user\_input.upcase!

puts user\_input.capitalize!

**34. Write a program to print numbers from 1 to n using until or while loop**

**Program:**

i = 1

while i <= 10 do

puts i

i += 1

end

k = 1

until k == 10 do

puts k

k += 1

end

**1.Create a file and write some text in the file using commandline**

fname = "sample.txt"

somefile = File.open(fname, "w")

somefile.puts "Hello file!"

somefile.close

**2.Read from the file**

file = File.open("sample.txt", "r")

contents = file.read

puts contents

contents = file.read

puts contents

sample output:

Hello file!

**3.Program that reads and print each line ?**

File.open("sample.txt").readlines.each do |line|

  puts line

end

Sample output:

Hello

File!

**4.Write a program to delete a file**

def remove\_file(sample.txt)

 File.delete(sample.txt)

end

**5.Program to convert Json file to xml**

require 'json'

require 'yaml'

json = '{ "x": "y" }'

data = JSON.parse(json)

yml = YAML::dump(data)

puts yml

**6.Write a program to parse yml**

require 'yaml' # STEP ONE, REQUIRE YAML!

# Parse a YAML string

YAML.load("--- foo")

YAML.dump("foo

{ :a => 'b'}.to\_yaml

**Sample output:**

"---\n:a: b\n"

7**.Write a program to convert array into YML using to\_yml**

require 'yaml'

names = %w[chris sandy josie billy suzie]

yaml\_example1 = names.to\_yaml

puts yaml\_example1

**Sampleoutput**:

---

- chris

- sandy

- josie

- billy

- suzie

=> nil

**8.Write a program that converts array to yml and loads the yml output and print array using load()**

require 'yaml'

names = %w[chris sandy josie billy suzie]

yaml\_example1 = names.to\_yaml

puts yaml\_example1

array\_example = YAML::load(yaml\_example2)

puts array\_example

**Sample output:**

---

- chris

- sandy

- josie

- billy

- suzie

chris

sandy

josie

billy

suzie

=> nil

**9.Write a program that converts array to yml using dump()**

require 'yaml’

names = %w[chris sandy josie billy suzie]

yaml\_example2 = YAML::dump(names)

puts yaml\_example2

---

- chris

- sandy

- josie

- billy

- suzie

=> nil

**10.Write a program parsing Json**

require 'json'

my\_hash = JSON.parse('{"hello": "goodbye"}')

puts my\_hash["hello"]

**Sample output:**

goodbye

=> nil

**11.Wrte a program that generates Json file**

require 'json'

my\_hash = {:hello => "goodbye"}

puts JSON.generate(my\_hash) => "{\"hello\":\"goodbye\"}"

**Sample Output:**

{"{\"hello\":\"goodbye\"}"=>"{\"hello\":\"goodbye\"}"}

=> nil

12.**Write a program converts array/string to json**

**#!/usr/bin/ruby**

**require 'rubygems'**

**require 'json'**

**require 'pp'**

**json = File.read('input.json')**

**obj = JSON.parse(json)**

**pp obj**

**{**

**..    "President"=>"Alan Isaac",**

**..    "CEO"=>"David Richardson",**

**..**

**..    "India"=>**

**..    ["Sachin Tendulkar", "Virender Sehwag", "Gautam Gambhir"],**

**13.Write a program to generate json using  pretty**

puts JSON.pretty\_generate([1, 2, {"a"=>3.141}, false, true, nil, 4..10])

[

 1,

 2,

 {

   "a": 3.141

 },

 false,

 true,

 nil,

 {

   "json\_class": "Range",

   "data": [

     4,

     10,

     false

   ]

 }

]

**Sampleoutput:**

=> [1, 2, {:a=>3.141}, false, true, nil, {:json\_class=>"Range", :data=>[4, 10, false]}]

**14.Write a program that connect to Database in your pc.**

**require 'rubygems'**

require 'DBI'

require 'pp'

datasource = 'oracle'

usr = 'scott'

pwd = 'oracleuser'

conn =

       DBI.connect("DBI:ODBC:#{datasource}", "#{usr}","#{pwd}")

puts conn.get\_server\_info

   rs = conn.query 'SELECT VERSION()'

   puts rs.fetch\_row

rescue Mysql::Error => e

   puts e.errno

   puts e.error

ensure

   con.close if con

end

**15.Write a program to perform a select query.**

require "sql"

begin

   con = Mysql.new 'localhost', 'scott', 'oracleuser'

   rs = con.query("SELECT \* FROM emp")

   n\_rows = rs.num\_rows

   puts "There are #{n\_rows} rows in the result set"

   n\_rows.times do

       puts rs.fetch\_row.join("\s")

   end

rescue Mysql::Error => e

   puts e.errno

   puts e.error

ensure

   con.close if con

end