

**SWE2034 – Ruby Programming**

Guided By – **Dr Yogesh C**

Slot – **L5+L6**

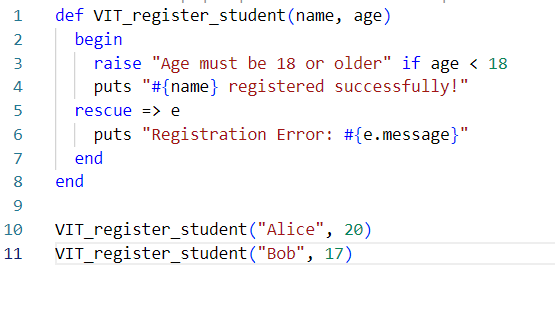
NAME: **VISHWANTH P**

REGISTER.NO: **21MIS1117**

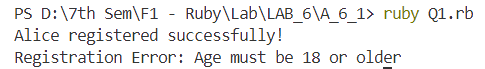
**Lab Assignment - 6**

**Assessment 6.1**

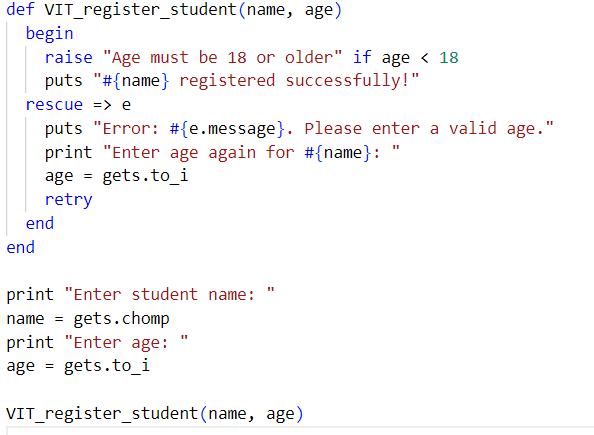
1. Implement exception handling using raise and rescue



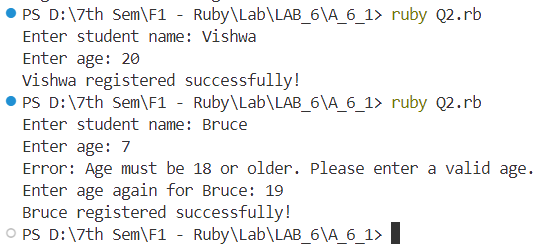
Output



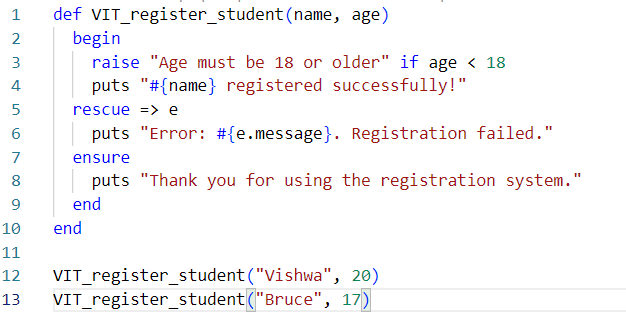
1. Implement exception handling using raise, rescue and retry



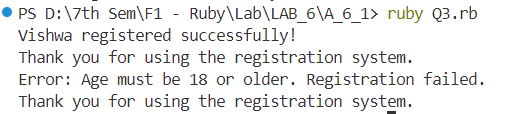
Output



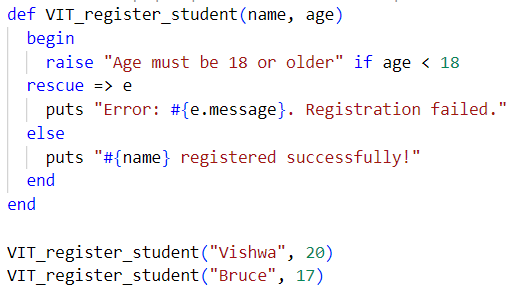
1. Implement exception handling using raise, rescue and ensure



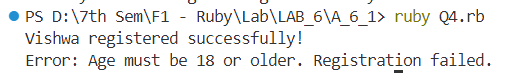
Output



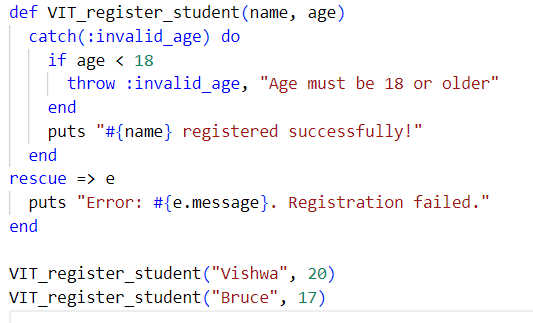
1. Implement exception handling using raise, rescue and else



Output



1. Implement Catch and Throw in Exception Handling



Output



**Assessment 6.2**

1. Implement Assertion methods, Benchmarking and profiling technique to measure the performance of any ruby code.

require 'minitest/autorun'

require 'benchmark'

require 'ruby-prof'

def VIT\_register\_student(name, age)

  if age < 18

    raise "Age must be 18 or older"

  else

    return "#{name} registered successfully!"

  end

end

class TestVITRegisterStudent < Minitest::Test

  def test\_VIT\_register\_student

    assert\_equal "Vishwa registered successfully!", VIT\_register\_student("Vishwa", 20), "The registration should succeed for Vishwa"

    assert\_equal "Bruce registered successfully!", VIT\_register\_student("Bruce", 19), "The registration should succeed for Bruce"

    assert\_raises(RuntimeError) { VIT\_register\_student("Dhoni", 17) }

  end

end

def benchmark\_VIT\_register\_student

  time = Benchmark.realtime do

    result = VIT\_register\_student("Vishwa", 20)

    puts result

  end

  puts "Benchmark: Registration time is #{time} seconds"

end

def profile\_VIT\_register\_student

  RubyProf.start

  VIT\_register\_student("Vishwa", 20)

  result = RubyProf.stop

  printer = RubyProf::FlatPrinter.new(result)

  printer.print(STDOUT)

end

puts "Running Assertions..."

Minitest.run

puts "\nBenchmarking VIT Register Student:"

benchmark\_VIT\_register\_student

puts "\nProfiling VIT Register Student:"

profile\_VIT\_register\_student

**Output**

PS D:\7th Sem\F1 - Ruby\Lab\LAB\_6\A\_6\_2> ruby Q1.rb

Running Assertions...

Run options: --seed 24486

# Running:

.

Finished in 0.001281s, 780.7620 runs/s, 2342.2861 assertions/s.

1 runs, 3 assertions, 0 failures, 0 errors, 0 skips

Benchmarking VIT Register Student:

Vishwa registered successfully!

Benchmark: Registration time is 0.00010339997243136168 seconds

Profiling VIT Register Student:

NOTE: RubyProf.start is deprecated; use RubyProf::Profile#start instead. It will be removed on or after 2023-06.

RubyProf.start called from Q1.rb:30.

NOTE: RubyProf.running? is deprecated; use RubyProf::Profile#running? instead. It will be removed on or after 2023-06.

RubyProf.running? called from C:/Users/Vishwanth Prakash/.local/share/gem/ruby/3.3.0/gems/ruby-prof-1.7.1-x64-mingw-ucrt/lib/ruby-prof/compatibility.rb:97.

NOTE: RubyProf.measure\_mode is deprecated; use RubyProf::Profile#measure\_mode instead. It will be removed on or after 2023-06.

RubyProf.measure\_mode called from C:/Users/Vishwanth Prakash/.local/share/gem/ruby/3.3.0/gems/ruby-prof-1.7.1-x64-mingw-ucrt/lib/ruby-prof/compatibility.rb:44.

NOTE: RubyProf.exclude\_threads is deprecated; use RubyProf::Profile#exclude\_threads instead. It will be removed on or after 2023-06.

RubyProf.exclude\_threads called from C:/Users/Vishwanth Prakash/.local/share/gem/ruby/3.3.0/gems/ruby-prof-1.7.1-x64-mingw-ucrt/lib/ruby-prof/compatibility.rb:44.

NOTE: RubyProf.stop is deprecated; use RubyProf::Profile#stop instead. It will be removed on or after 2023-06.

RubyProf.stop called from Q1.rb:32.

NOTE: RubyProf.running? is deprecated; use RubyProf::Profile#running? instead. It will be removed on or after 2023-06.

RubyProf.running? called from C:/Users/Vishwanth Prakash/.local/share/gem/ruby/3.3.0/gems/ruby-prof-1.7.1-x64-mingw-ucrt/lib/ruby-prof/compatibility.rb:93.

Measure Mode: wall\_time

Thread ID: 440

Fiber ID: 420

Total: 0.000386

Sort by: self\_time

%self total self wait child calls name location

79.97 0.000 0.000 0.000 0.000 2 Warning#warn

9.64 0.000 0.000 0.000 0.000 1 Object#profile\_VIT\_register\_student Q1.rb:31

2.07 0.000 0.000 0.000 0.000 2 Kernel#caller

1.79 0.000 0.000 0.000 0.000 2 <Module::Gem>#location\_of\_caller C:/Ruby33-x64/lib/ruby/3.3.0/rubygems.rb:626

1.68 0.000 0.000 0.000 0.000 2 String#=~

1.17 0.000 0.000 0.000 0.000 4 Array#join

0.75 0.000 0.000 0.000 0.000 1 Object#VIT\_register\_student Q1.rb:5

0.57 0.000 0.000 0.000 0.000 2 <Module::Gem::Deprecate>#skip C:/Ruby33-x64/lib/ruby/3.3.0/rubygems/deprecate.rb:74

0.54 0.000 0.000 0.000 0.000 2 String#to\_i

0.54 0.000 0.000 0.000 0.000 2 String#==

0.39 0.000 0.000 0.000 0.000 2 Warning::buffer#write

0.36 0.000 0.000 0.000 0.000 2 Integer#to\_s

0.28 0.000 0.000 0.000 0.000 1 Integer#<

0.23 0.000 0.000 0.000 0.000 2 Array#[]

\* recursively called methods

Columns are:

%self - The percentage of time spent in this method, derived from self\_time/total\_time.

total - The time spent in this method and its children.

self - The time spent in this method.

wait - The amount of time this method waited for other threads.

child - The time spent in this method's children.

calls - The number of times this method was called.

name - The name of the method.

location - The location of the method.

The interpretation of method names is:

\* MyObject#test - An instance method "test" of the class "MyObject"

\* <Object:MyObject>#test - The <> characters indicate a method on a singleton class.

\* <Object:MyObject>#test - The <> characters indicate a method on a singleton class.

\* <Object:MyObject>#test - The <> characters indicate a method on a singleton class.

Run options: --seed 47163

# Running:

.

\* <Object:MyObject>#test - The <> characters indicate a method on a singleton class.

Run options: --seed 47163

# Running:

.

\* <Object:MyObject>#test - The <> characters indicate a method on a singleton class.

Run options: --seed 47163

# Running:

.

\* <Object:MyObject>#test - The <> characters indicate a method on a singleton class.

Run options: --seed 47163

\* <Object:MyObject>#test - The <> characters indicate a method on a singleton class.

Run options: --seed 47163

\* <Object:MyObject>#test - The <> characters indicate a method on a singleton class.

Run options: --seed 47163

# Running:

\* <Object:MyObject>#test - The <> characters indicate a method on a singleton class.

Run options: --seed 47163

\* <Object:MyObject>#test - The <> characters indicate a method on a singleton class.

\* <Object:MyObject>#test - The <> characters indicate a method on a singleton class.

\* <Object:MyObject>#test - The <> characters indicate a method on a singleton class.

\* <Object:MyObject>#test - The <> characters indicate a method on a singleton class.

Run options: --seed 47163

# Running:

.

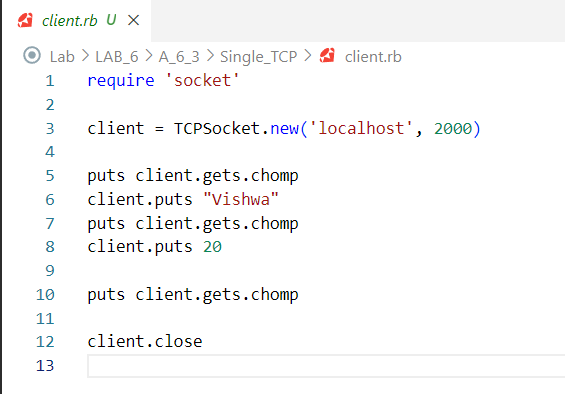
Finished in 0.000899s, 1112.0997 runs/s, 3336.2990 assertions/s.

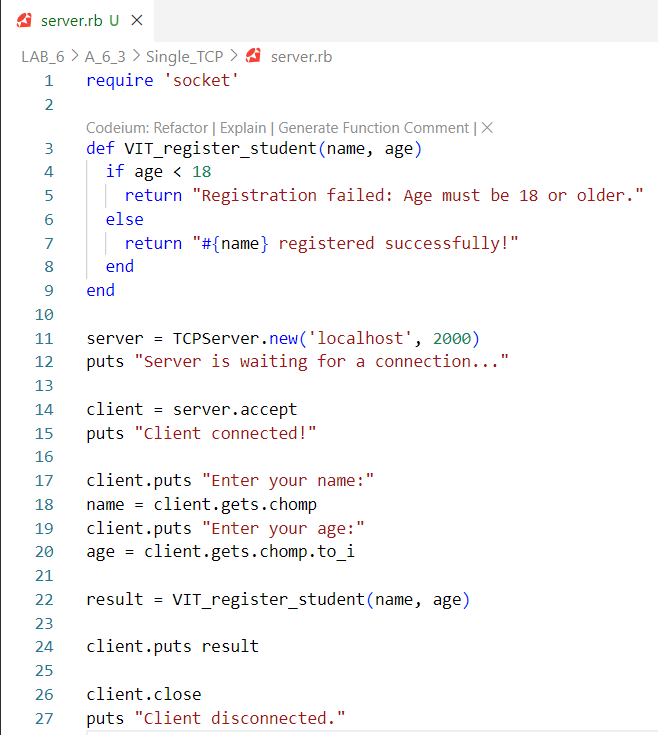
1 runs, 3 assertions, 0 failures, 0 errors, 0 skips

**Assessment 6.3**

1. Implement Single and Multiple TCP Connections

**Single TCP Connection**

****

****

**Server Output**

Server is waiting for a connection...

Client connected!

Client disconnected.

**Client Output**

Enter your name:

Enter your age:

Vishwa

20

Vishwa registered successfully!

**Multiple TCP Connection**

**Client.rb**

require 'socket'

client = TCPSocket.new('localhost', 2000)

puts client.gets.chomp

client.puts "Vishwa"

puts client.gets.chomp

client.puts 20

puts client.gets.chomp

client.close

puts "Connection closed."

**Server.rb**

require 'socket'

require 'thread'

def VIT\_register\_student(name, age)

  if age < 18

    return "Registration failed: Age must be 18 or older."

  else

    return "#{name} registered successfully!"

  end

end

server = TCPServer.new('localhost', 2000)

puts "Server is waiting for connections..."

loop do

  client = server.accept

  Thread.new(client) do |client|

    client.puts "Enter your name:"

    name = client.gets.chomp

    client.puts "Enter your age:"

    age = client.gets.chomp.to\_i

    result = VIT\_register\_student(name, age)

    client.puts result

    client.close

    puts "Client disconnected."

  end

end

**Server Output**

Server is waiting for connections...

Client connected!

Received name: Vishwa

Received age: 20

Sent result to client: Vishwa registered successfully!

Client disconnected.

**Client Output**

Enter your name:

Enter your age:

Vishwa

20

Vishwa registered successfully!

Connection closed.