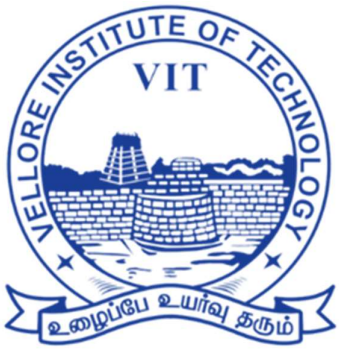

VIT®

CHENNAI

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

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

```ruby
def VIT_register_student(name, age)
  begin
    raise "Age must be 18 or older" if age < 18
    puts "#{name} registered successfully!"
  rescue => e
    puts "Registration Error: #{e.message}"
  end
end

VIT_register_student("Alice", 20)
VIT_register_student("Bob", 17)
```

Output

```
PS D:\7th Sem\F1 - Ruby\Lab\LAB_6\A_6_1> ruby Q1.rb
Alice registered successfully!
Registration Error: Age must be 18 or older
```

2) Implement exception handling using raise, rescue and retry

```ruby
def VIT_register_student(name, age)
  begin
    raise "Age must be 18 or older" if age < 18
    puts "#{name} registered successfully!"
  rescue => e
    puts "Error: #{e.message}. Please enter a valid age."
    print "Enter age again for #{name}: "
    age = gets.to_i
    retry
  end
end

print "Enter student name: "
name = gets.chomp
print "Enter age: "
age = gets.to_i

VIT_register_student(name, age)
```

Output

```
PS D:\7th Sem\F1 - Ruby\Lab\LAB_6\A_6_1> ruby Q2.rb
Enter student name: Vishwa
Enter age: 20
Vishwa registered successfully!
PS D:\7th Sem\F1 - Ruby\Lab\LAB_6\A_6_1> ruby Q2.rb
Enter student name: Bruce
Enter age: 7
Error: Age must be 18 or older. Please enter a valid age.
Enter age again for Bruce: 19
Bruce registered successfully!
PS D:\7th Sem\F1 - Ruby\Lab\LAB_6\A_6_1>
```

3) Implement exception handling using raise, rescue and ensure

```ruby
def VIT_register_student(name, age)
  begin
    raise "Age must be 18 or older" if age < 18
    puts "#{name} registered successfully!"
  rescue => e
    puts "Error: #{e.message}. Registration failed."
  ensure
    puts "Thank you for using the registration system."
  end
end

VIT_register_student("Vishwa", 20)
VIT_register_student("Bruce", 17)
```

Output

```
PS D:\7th Sem\F1 - Ruby\Lab\LAB_6\A_6_1> ruby Q3.rb
Vishwa registered successfully!
Thank you for using the registration system.
Error: Age must be 18 or older. Registration failed.
Thank you for using the registration system.
```

4) Implement exception handling using raise, rescue and else

```ruby
def VIT_register_student(name, age)
  begin
    raise "Age must be 18 or older" if age < 18
  rescue => e
    puts "Error: #{e.message}. Registration failed."
  else
    puts "#{name} registered successfully!"
  end
end

VIT_register_student("Vishwa", 20)
VIT_register_student("Bruce", 17)
```

Output

```
PS D:\7th Sem\F1 - Ruby\Lab\LAB_6\A_6_1> ruby Q4.rb
Vishwa registered successfully!
Error: Age must be 18 or older. Registration failed.
```

5) Implement Catch and Throw in Exception Handling

```ruby
def VIT_register_student(name, age)
  catch(:invalid_age) do
    if age < 18
      throw :invalid_age, "Age must be 18 or older"
    end
    puts "#{name} registered successfully!"
  end
rescue => e
  puts "Error: #{e.message}. Registration failed."
end

VIT_register_student("Vishwa", 20)
VIT_register_student("Bruce", 17)
```

Output

```
PS D:\7th Sem\F1 - Ruby\Lab\LAB_6\A_6_1> ruby Q5.rb
Vishwa registered successfully!
```

Assessment 6.2

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

```ruby
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

7) Implement Single and Multiple TCP Connections

Single TCP Connection

client.rb U

Lab > LAB_6 > A_6_3 > Single_TCP > client.rb

```ruby
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

```

server.rb U

LAB_6 > A_6_3 > Single_TCP > server.rb

```ruby
require 'socket'

Codeium: Refactor | Explain | Generate Function Comment |
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 a connection..."

client = server.accept
puts "Client connected!"

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."
```

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

```ruby
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

```ruby
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.
```