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 Assessment - 1

1. Introduction

a. Variables and Methods

i. Different Types of Variables and Scope

Local Variable

21MIS1117_LAB1.rb

```ruby
def define_Local
  local_var = "local variable"
  puts local_var
end

define_Local
```

```
PS D:\7th Sem\F1 - Ruby\Lab> ruby 21MIS1117_LAB1.rb
local variable
PS D:\7th Sem\F1 - Ruby\Lab>
```

Instance Variable

21MIS1117_LAB1.rb

```ruby
class ExampleClass
  def initialize(name)
    @name = name
  end

  def display_name
    puts "Name is #{@name}"
  end
end

example = ExampleClass.new("Vishwanth")
example.display_name
```

```
PS D:\7th Sem\F1 - Ruby\Lab> ruby 21MIS1117_LAB1.rb
Name is Vishwanth
PS D:\7th Sem\F1 - Ruby\Lab>
```

Class Variable

21MIS1117_LAB1.rb

```ruby
class ExampleClass
  @@class_var = 0

  def initialize(name)
    @name = name
    @@class_var += 1
  end

  def display_count
    puts "Number of instances: #{@@class_var}"
  end
end

example1 = ExampleClass.new("Vishwanth")
example2 = ExampleClass.new("Prakash")
example1.display_count
example2.display_count
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE POLYGLOT NOTEBOOK

TERMINAL powershell

```
PS D:\7th Sem\F1 - Ruby\Lab> ruby 21MIS1117_LAB1.rb
Number of instances: 2
Number of instances: 2
PS D:\7th Sem\F1 - Ruby\Lab>
```

Global Variables

21MIS1117_LAB1.rb

```ruby
$global_var = "global variable"

def example_method
  puts $global_var
end

example_method

$global_var = "Global variable modified"
puts $global_var
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE POLYGLOT NOTEBOOK

TERMINAL

```
PS D:\7th Sem\F1 - Ruby\Lab> ruby 21MIS1117_LAB1.rb
global variable
Global variable modified
PS D:\7th Sem\F1 - Ruby\Lab>
```

Constants

```
21MIS1117_LAB1.rb
CONSTANT_VAR = "constant variable"

Codeium: Refactor | Explain | Generate Function Comment |
def example_method
  puts CONSTANT_VAR
end

example_method

CONSTANT_VAR = "Constant variable modified"
puts CONSTANT_VAR
```

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   POSTMAN CONSOLE   POLYGLOT NOTEBOO
TERMINAL                                                  powershell
PS D:\7th Sem\F1 - Ruby\Lab> ruby 21MIS1117_LAB1.rb
constant variable
21MIS1117_LAB1.rb:9: warning: already initialized constant CONSTANT_VAR
21MIS1117_LAB1.rb:1: warning: previous definition of CONSTANT_VAR was here
Constant variable modified
PS D:\7th Sem\F1 - Ruby\Lab>
```

ii. Arithmetic, Relation, Logical and Boolean operators (Each 2 Programs)

Arithmetic Operators

Program 1: Basic Arithmetic Operations

```
21MIS1117_LAB1.rb
a = 10
b = 5
puts "Addition: #{a} + #{b} = #{a + b}"

puts "Subtraction: #{a} - #{b} = #{a - b}"

puts "Multiplication: #{a} * #{b} = #{a * b}"

puts "Division: #{a} / #{b} = #{a / b}"
```

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   POSTMAN CONSOLE   POLYGLOT NOTEBOOK
TERMINAL                                                  powershell
PS D:\7th Sem\F1 - Ruby\Lab> ruby 21MIS1117_LAB1.rb
Addition: 10 + 5 = 15
Subtraction: 10 - 5 = 5
Multiplication: 10 * 5 = 50
Division: 10 / 5 = 2
PS D:\7th Sem\F1 - Ruby\Lab>
```

Program 2: Modulus and Exponentiation

21MIS1117_LAB1.rb

```ruby
a = 10
b = 3
puts "Modulus: #{a} % #{b} = #{a % b}"

puts "Exponentiation: #{a} ** #{b} = #{a ** b}"
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE POLYGLOT NOTEBOOK

TERMINAL powershell

```
PS D:\7th Sem\F1 - Ruby\Lab> ruby 21MIS1117_LAB1.rb
Modulus: 10 % 3 = 1
Exponentiation: 10 ** 3 = 1000
PS D:\7th Sem\F1 - Ruby\Lab>
```

Relational Operators

Program 1: Basic Comparisons

21MIS1117_LAB1.rb

```ruby
a = 10
b = 5
puts "Is #{a} greater than #{b}? #{a > b}"
puts "Is #{a} less than #{b}? #{a < b}"
puts "Is #{a} equal to #{b}? #{a == b}"
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE POLYGLOT NOTEBOO

TERMINAL powershell

```
PS D:\7th Sem\F1 - Ruby\Lab> ruby 21MIS1117_LAB1.rb
Is 10 greater than 5? true
Is 10 less than 5? false
Is 10 equal to 5? false
PS D:\7th Sem\F1 - Ruby\Lab>
```

Program 2: Complex Comparisons

21MIS1117_LAB1.rb

```ruby
a = 10
b = 10
c = 5

puts "Is #{a} greater than or equal to #{b}? #{a >= b}"
puts "Is #{c} less than or equal to #{b}? #{c <= b}"
puts "Is #{a} not equal to #{c}? #{a != c}"
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE POLYGLOT NOTEBOO

TERMINAL powershell

```
PS D:\7th Sem\F1 - Ruby\Lab> ruby 21MIS1117_LAB1.rb
Is 10 greater than or equal to 10? true
Is 5 less than or equal to 10? true
Is 10 not equal to 5? true
PS D:\7th Sem\F1 - Ruby\Lab>
```

Logical Operators

Program 1: Logical AND and OR

21MIS1117_LAB1.rb

```ruby
a = true
b = false

puts "a AND b: #{a && b}"
puts "a OR b: #{a || b}"
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE POLYGLOT NOTEBOOK

TERMINAL powershell

```
PS D:\7th Sem\F1 - Ruby\Lab> ruby 21MIS1117_LAB1.rb
a AND b: false
a OR b: true
PS D:\7th Sem\F1 - Ruby\Lab>
```

Program 2: Logical NOT

21MIS1117_LAB1.rb

```ruby
a = true
b = false

puts "NOT a: #{!a}"
puts "NOT b: #{!b}"
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE POLYGLOT NOTEBOOK

TERMINAL powershell

```
PS D:\7th Sem\F1 - Ruby\Lab> ruby 21MIS1117_LAB1.rb
NOT a: false
NOT b: true
PS D:\7th Sem\F1 - Ruby\Lab>
```

Boolean Operators

Program 1: Combining Conditions

21MIS1117_LAB1.rb

```ruby
age = 25
has_license = true

can_drive = age >= 18 && has_license
puts "Can drive? #{can_drive}"
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE POLYGLOT NOTEBOOK

TERMINAL powershell

```
PS D:\7th Sem\F1 - Ruby\Lab> ruby 21MIS1117_LAB1.rb
Can drive? true
PS D:\7th Sem\F1 - Ruby\Lab>
```

Program 2: Using OR to Provide Alternatives

21MIS1117_LAB1.rb

```ruby
is_weekend = true
is_holiday = false

can_relax = is_weekend || is_holiday
puts "Can relax? #{can_relax}"
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE POLYGLOT NOTEBOOK

TERMINAL powershell

```
PS D:\7th Sem\F1 - Ruby\Lab> ruby 21MIS1117_LAB1.rb
Can relax? true
PS D:\7th Sem\F1 - Ruby\Lab>
```

b. Conversion of Variables

21MIS1117_LAB1.rb

```ruby
str_num = "123"
str_float = "123.45"
num = 123
array = [[:key1, "value1"], [:key2, "value2"]]
hash = {key1: "value1", key2: "value2"}

int_val = str_num.to_i
puts "String '#{str_num}' to Integer: #{int_val} (#{int_val.class})"

float_val = str_float.to_f
puts "String '#{str_float}' to Float: #{float_val} (#{float_val.class})"

str_from_int = num.to_s
puts "Integer #{num} to String: '#{str_from_int}' (#{str_from_int.class})"

symbol = str_num.to_sym
puts "String '#{str_num}' to Symbol: :#{symbol} (#{symbol.class})"

hash_from_array = array.to_h
puts "Array #{array} to Hash: #{hash_from_array} (#{hash_from_array.class})"

array_from_hash = hash.to_a
puts "Hash #{hash} to Array: #{array_from_hash} (#{array_from_hash.class})"

array = [1, 2, 3]
str_from_array = array.join(", ")
new_array = str_from_array.split(", ").map(&:to_i)
puts "Array #{array} to String '#{str_from_array}' and back to Array #{new_array} (#{new_array.class})"
```

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   POSTMAN CONSOLE   POLYGLOT NOTEBOOK   SERIAL MONITOR
TERMINAL                                                                                  powershell
PS D:\7th Sem\F1 - Ruby\Lab> ruby 21MIS1117_LAB1.rb
String '123' to Integer: 123 (Integer)
String '123.45' to Float: 123.45 (Float)
Integer 123 to String: '123' (String)
String '123' to Symbol: :123 (Symbol)
Array [[:key1, "value1"], [:key2, "value2"]] to Hash: {:key1=>"value1", :key2=>"value2"} (Hash)
Hash {:key1=>"value1", :key2=>"value2"} to Array: [[:key1, "value1"], [:key2, "value2"]] (Array)
Array [1, 2, 3] to String '1, 2, 3' and back to Array [1, 2, 3] (Array)
PS D:\7th Sem\F1 - Ruby\Lab>
```

c. Inclusive and Non-Inclusive Ranges

```
21MIS1117_LAB1.rb ×
21MIS1117_LAB1.rb
1   inclusive_range = 1..5
2   puts "Inclusive Range (1..5):"
3   inclusive_range.each do |num|
4     print "#{num} "
5   end
6   puts "\nDoes the inclusive range include 5? #{inclusive_range.include?(5)}"
7
8   non_inclusive_range = 1...5
9   puts "\nNon-Inclusive Range (1...5):"
10  non_inclusive_range.each do |num|
11    print "#{num} "
12  end
13  puts "\nDoes the non-inclusive range include 5? #{non_inclusive_range.include?(5)}"
14
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   POSTMAN CONSOLE   POLYGLOT NOTEBOOK   SERIAL MONITOR
TERMINAL                                                                                  powershell
PS D:\7th Sem\F1 - Ruby\Lab> ruby 21MIS1117_LAB1.rb
Inclusive Range (1..5):
1 2 3 4 5
Does the inclusive range include 5? true

Non-Inclusive Range (1...5):
1 2 3 4
Does the non-inclusive range include 5? false
PS D:\7th Sem\F1 - Ruby\Lab>
```

d. Redo Program

21MIS1117_LAB1.rb

```ruby
loop do
  puts "Please enter a number between 1 and 5:"
  input = gets.chomp.to_i

  if input >= 1 && input <= 5
    puts "Thank you! You entered a valid number: #{input}"
    break
  else
    puts "Invalid input. Please try again."
    redo
  end
end
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE POLYGLOT NOTEBOOK SERIAL MONITOR

TERMINAL powershell

```
PS D:\7th Sem\F1 - Ruby\Lab> ruby 21MIS1117_LAB1.rb
Please enter a number between 1 and 5:
4
Thank you! You entered a valid number: 4
PS D:\7th Sem\F1 - Ruby\Lab> ruby 21MIS1117_LAB1.rb
Please enter a number between 1 and 5:
8
Invalid input. Please try again.
Please enter a number between 1 and 5:
6
Invalid input. Please try again.
Please enter a number between 1 and 5:
1
Thank you! You entered a valid number: 1
PS D:\7th Sem\F1 - Ruby\Lab>
```

e. Break Program

21MIS1117_LAB1.rb

```ruby
secret_number = 7

loop do
  puts "Guess the secret number (between 1 and 10):"
  guess = gets.chomp.to_i

  if guess == secret_number
    puts "Congratulations! You guessed the correct number: #{secret_number}"
    break
  else
    puts "Wrong guess. Try again."
  end
end

puts "Game over."
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE POLYGLOT NOTEBOOK SERIAL MONITOR

TERMINAL powershell

```
PS D:\7th Sem\F1 - Ruby\Lab> ruby 21MIS1117_LAB1.rb
Guess the secret number (between 1 and 10):
9
Wrong guess. Try again.
Guess the secret number (between 1 and 10):
4
Wrong guess. Try again.
Guess the secret number (between 1 and 10):
7
Congratulations! You guessed the correct number: 7
Game over.
PS D:\7th Sem\F1 - Ruby\Lab>
```

2. A program that prints out the first 10 even numbers

21MIS1117_LAB1.rb

```ruby
even_count = 0
number = 1

while even_count < 10
  if number.even?
    puts number
    even_count += 1
  end
  number += 1
end
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

TERMINAL

```
PS D:\7th Sem\F1 - Ruby\Lab> ruby 21MIS1117_LAB1.rb
2
4
6
8
10
12
14
16
18
20
PS D:\7th Sem\F1 - Ruby\Lab>
```

3. A program that calculates the factorial of a number using a while loop

21MIS1117_LAB1.rb

```ruby
fact = 1
num = 7
i=num

while i>0
  fact = fact*i
  i-=1
end
puts fact
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE POLYG

TERMINAL powershell

```
PS D:\7th Sem\F1 - Ruby\Lab> ruby 21MIS1117_LAB1.rb
5040
PS D:\7th Sem\F1 - Ruby\Lab>
```

4. A program that prints out the sum of all numbers from 1 to 100 using a for loop S.

21MIS1117_LAB1.rb

```ruby
sum = 0

for number in 1..100
  sum += number
end

puts "The sum of all numbers from 1 to 100 is #{sum}."
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE PO

TERMINAL

```
PS D:\7th Sem\F1 - Ruby\Lab> ruby 21MIS1117_LAB1.rb
The sum of all numbers from 1 to 100 is 5050.
PS D:\7th Sem\F1 - Ruby\Lab>
```

5. A program that prints out the multiplication table of a number

21MIS1117_LAB1.rb

```ruby
puts "Enter a number to generate its multiplication table:"
number = gets.chomp.to_i

range = 1..10

puts "Multiplication table for #{number}:"
for i in range
  puts "#{number} x #{i} = #{number * i}"
end
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE POLYGLOT NOTEBOOK SERIAL MONITOR

TERMINAL powershell

```
PS D:\7th Sem\F1 - Ruby\Lab> ruby 21MIS1117_LAB1.rb
Enter a number to generate its multiplication table:
6
Multiplication table for 6:
6 x 1 = 6
6 x 2 = 12
6 x 3 = 18
6 x 4 = 24
6 x 5 = 30
6 x 6 = 36
6 x 7 = 42
6 x 8 = 48
6 x 9 = 54
6 x 10 = 60
PS D:\7th Sem\F1 - Ruby\Lab>
```

6. A program that checks if a number is prime or not

21MIS1117_LAB1.rb

```ruby
def prime?(number)
  return false if number < 2

  (2..Math.sqrt(number)).each do |i|
    return false if number % i == 0
  end

  true
end

puts "Enter a number to check if it is prime:"
number = gets.chomp.to_i

if prime?(number)
  puts "#{number} is a prime number."
else
  puts "#{number} is not a prime number."
end
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE

TERMINAL

```
PS D:\7th Sem\F1 - Ruby\Lab> ruby 21MIS1117_LAB1.rb
Enter a number to check if it is prime:
50
50 is not a prime number.
PS D:\7th Sem\F1 - Ruby\Lab> ruby 21MIS1117_LAB1.rb
Enter a number to check if it is prime:
19
19 is a prime number.
PS D:\7th Sem\F1 - Ruby\Lab>
```

7. A program that prints out the Fibonacci sequence up to a certain number

21MIS1117_LAB1.rb

```ruby
f_n = 0
s_n = 1
10.times do
  puts f_n
  f_n, s_n = s_n, f_n+s_n
end
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE POLYGL

TERMINAL powershell

```
PS D:\7th Sem\F1 - Ruby\Lab> ruby 21MIS1117_LAB1.rb
0
1
1
2
3
5
8
13
21
34
PS D:\7th Sem\F1 - Ruby\Lab>
```

8. A program that prints out a triangle of stars

21MIS1117_LAB1.rb

```ruby
puts "Enter the number of rows for the triangle:"
rows = gets.chomp.to_i

(1..rows).each do |i|
  puts "*" * i
end
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE POLYG

TERMINAL

```
PS D:\7th Sem\F1 - Ruby\Lab> ruby 21MIS1117_LAB1.rb
Enter the number of rows for the triangle:
9
*
**
***
****
*****
******
*******
********
*********
PS D:\7th Sem\F1 - Ruby\Lab>
```

9. A program that reverses a string using a while loop

21MIS1117_LAB1.rb

```ruby
puts "Enter a string to reverse:"
input_string = gets.chomp

reversed_string = ""

index = input_string.length - 1

while index >= 0
  reversed_string += input_string[index]
  index -= 1
end

puts "Reversed string: #{reversed_string}"
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE POLYGL

TERMINAL powershell

```
PS D:\7th Sem\F1 - Ruby\Lab> ruby 21MIS1117_LAB1.rb
Enter a string to reverse:
Vishwanth Prakash
Reversed string: hsakarP htnawhsiV
PS D:\7th Sem\F1 - Ruby\Lab>
```

10. A program that prints out the comm-elements between two arrays

21MIS1117_LAB1.rb

```ruby
array1 = [1, 2, 3, 4, 5]

array2 = [4, 5, 6, 7, 8]

common_elements = array1 & array2

puts "Common elements between the two arrays are: #{common_elements}"
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE POLYGLOT NOT

TERMINAL powershell

```
PS D:\7th Sem\F1 - Ruby\Lab> ruby 21MIS1117_LAB1.rb
Common elements between the two arrays are: [4, 5]
PS D:\7th Sem\F1 - Ruby\Lab>
```