UIT2602 WEB PROGRAMMING
Exercise 3: Ruby Programming

Name: S. Selcia
Reg. No.: 3122215002098
Section: IT B

1. **Aim:**

To practice various Ruby programming concepts including basic syntax, control structures, user input, functions, arrays, hashes, and file handling.

2. **Required Web Tools and Methodology:**

- Ruby Programming Language
- Text Editor
- Terminal or command prompt

3. **Implementation Procedure:**

A. Basic Syntax Practice:
   a. Print "Hello, World!" to the console.
   b. Declare and manipulate variables of different types (string, integer, float, array, hash).

B. Control Structures:
   a. Implement conditional statements (`if-else`, `case`) for simple decision making.
   b. Use loops (`while`, `for`) for repetitive tasks.
   c. Try nested loops and conditional statements for more complex logic.

C. Get User Input:
   a. Write a Ruby script to get user input for various purposes.

D. Functions and Methods:
   a. Define and call functions with parameters.
   b. Explore default parameter values and variable-length argument lists.
   c. Experiment with returning values from functions.

E. Arrays and Enumerable:
   a. Create arrays and manipulate them (add elements, remove elements, access elements).
   b. Use enumerable methods like `map`, `select`, `reduce` for array manipulation.

F. File Handling:
   a. Write a Ruby script that interacts with files to perform operations such as reading, writing, and manipulating file contents.

4. Code Snippets

basic_syntax.rb

```ruby
puts "---------------------------------------------------------"
puts "1. Write a program to print 'Hello, World!' to the console."
puts "---------------------------------------------------------"
puts("Hello world")
puts "---------------------------------------------------------"
puts "2. Practice variable declaration and manipulation."
puts "---------------------------------------------------------"
# String variable
name = "Selcia"
puts "Hello, #{name}!"
# Integer variable
age = 30
puts "Age: #{age}"
# Float variable
height = 5.8
puts "Height: #{height}"
# Array variable
fruits = ["apple", "banana", "cherry"]
puts "Fruits: #{fruits}"
# Hash variable
person = { name: "Bob", age: 25, city: "New York" }
puts "Person: #{person}"
puts "----------------------------------------------------------------------------------------"
puts "3. Explore different data types such as integers, floats, strings, arrays, and hashes. "
puts "----------------------------------------------------------------------------------------"
# Integers
num1 = 10
num2 = 20
sum = num1 + num2
puts "Sum: #{sum}"
# Floats
pi = 3.14
radius = 5
area = pi * radius * radius
puts "Area of circle: #{area}"
# Strings
greeting = "Hello"
name = "Alice"
full_greeting = "#{greeting}, #{name}!"
puts full_greeting
# Arrays
numbers = [1, 2, 3, 4, 5]
puts "First element: #{numbers.first}"
puts "Last element: #{numbers.last}"
# Hashes
person = { name: "Alice", age: 30, city: "London" }
puts "Person's name: #{person[:name]}"
```

control_structures.rb

```ruby
puts "----------------------------------------------------------------------------"
puts "Implement conditional statements (if-else, case) for simple decision making."
puts "----------------------------------------------------------------------------"
# If-else statement
puts "Enter your age:"
age = gets.chomp.to_i

if age >= 18
  puts "Your age is #{age}, You are eligible to vote"
else
  puts "Your age is #{age}, You are not eligible to vote"
end

# Case statement
puts "Enter a day of the week:"
day = gets.chomp

case day
when "Monday"
  puts "It's Monday"
when "Tuesday"
  puts "It's Tuesday"
else
  puts "It's neither Monday nor Tuesday"
end

puts "----------------------------------------------------------------------------"
puts "Use loops (while, for) for repetitive tasks. "
puts "----------------------------------------------------------------------------"

# While loop
i = 1
while i <= 5
  puts "Iteration #{i}"
  i += 1
end

# For loop
for i in 1..5
  puts "Iteration #{i}"
end

puts "----------------------------------------------------------------------------"
puts "Try nested loops and conditional statements for more complex logic. "
puts "----------------------------------------------------------------------------"

# Nested loops
for i in 1..3
  for j in 1..3
    puts "#{i}, #{j}"
  end
end
```

```
# Nested conditional statements
puts "Enter temperature: "
temperature = gets.chomp.to_i
puts "Enter time of the day (morning, etc): "
time_of_day = gets.chomp
if temperature > 20
  if time_of_day == "morning"
    puts "It's a warm morning"
  else
    puts "It's a warm day"
  end
else
  puts "It's not very warm"
end
```

functions_methods.rb

```
puts "----------------------------------------------------------------------------"
puts "Define and call functions with parameters"
puts "----------------------------------------------------------------------------"

# Define a function with parameters
def greet(name)
  puts "Hello, #{name}!"
end

# Call the function with an argument
greet("Alice")

puts "----------------------------------------------------------------------------"
puts "Explore default parameter values and variable-length argument lists."
puts "----------------------------------------------------------------------------"

# Define a function with a default parameter value
def greet(name = "Anonymous")
  puts "Hello, #{name}!"
end

# Call the function without providing an argument
greet

# Define a function with a variable-length argument list
def sum(*numbers)
  total = 0
  numbers.each { |num| total += num }
  total
end

# Call the function with different numbers of arguments
puts sum(1, 2, 3)
puts sum(4, 5, 6, 7)

puts "----------------------------------------------------------------------------"
```

```ruby
puts "Experiment with returning values from functions."
puts "---------------------------------------------------------------------------"

# Define a function that returns a value
def square(number)
  number * number
end

# Call the function and use the returned value
result = square(5)
puts "Square of 5 is: #{result}"
```

arrays_enumerable.rb

```ruby
puts "------------------------------------------------------------------------------------"
puts "Create arrays and manipulate them (add elements, remove elements, access elements). "
puts "------------------------------------------------------------------------------------"

# Create an empty array
numbers = []
puts "Array: #{numbers}"
# Add elements to the array
numbers.push(1)
puts "Array: #{numbers}"
numbers << 2
puts "Array: #{numbers}"
numbers.unshift(0)   # add an element to the beginning of an array
puts "Array after adding elements: #{numbers}"

# Remove elements from the array
numbers.pop
puts "Array: #{numbers}"
numbers.shift    # remove and return the first element of an array
puts "Array: #{numbers}"
puts "Array after removing elements: #{numbers}"

# Access elements in the array
puts "First element: #{numbers.first}"
puts "Last element: #{numbers.last}"

puts "------------------------------------------------------------------------------------"
puts "Use enumerable methods like map, select, reduce for array manipulation. "
puts "------------------------------------------------------------------------------------"
# Create an array of numbers
numbers = [1, 2, 3, 4, 5]

# Map: Multiply each element by 2
doubled_numbers = numbers.map { |num| num * 2 }
puts "Doubled numbers: #{doubled_numbers}"

# Select: Select even numbers
even_numbers = numbers.select { |num| num.even? }
puts "Even numbers: #{even_numbers}"
```

```ruby
# Reduce: Calculate the sum of all elements
sum = numbers.reduce(0) { |acc, num| acc + num }
puts "Sum of numbers: #{sum}"
```

ex1.rb

```ruby
def copy_string(str, n)
  if n < 0
    return "Invalid input: n must be a non-negative integer."
  end

  result = ""
  n.times do
    result += str
  end
  return result
end

# Get user input
puts "Enter a string:"
input_str = gets.chomp

puts "Enter a non-negative integer:"
input_n = gets.chomp.to_i

# Call the function and print the result
result = copy_string(input_str, input_n)
puts "Result: #{result}"
```

ex2.rb

```ruby
def get_file_extension(filename)
  # Split the filename by '.' and get the last element
  parts = filename.split(".")
  if parts.length > 1
    return parts.last
  else
    return "No extension found."
  end
end

# Get user input
puts "Enter a filename:"
filename = gets.chomp

# Call the function and print the result
extension = get_file_extension(filename)
puts "Extension of the file: #{extension}"
```

ex3.rb

```ruby
# Get user input
puts "Enter your first name:"
first_name = gets.chomp

puts "Enter your last name:"
last_name = gets.chomp

# Concatenate the first and last names, then reverse the string
full_name = "#{first_name} #{last_name}"
reversed_name = full_name.reverse

# Print the reversed name
puts "Your name in reverse order is: #{reversed_name}"
```

ex4.rb

```ruby
def calculate_total_marks(subject_marks)
  total_marks = 0
  subject_marks.each do |subject, marks|
    total_marks += marks
  end
  return total_marks
end

# Hash containing subject names and marks
subject_marks = {
  "Maths" => 95,
  "Science" => 87,
  "English" => 92
}

# Calculate total marks
total_marks = calculate_total_marks(subject_marks)

# Print total marks
puts "Total marks: #{total_marks}"
```

ex5.rb

```ruby
def check_temperatures(temp1, temp2)
  if (temp1 < 0 && temp2 > 100) || (temp1 > 100 && temp2 < 0)
    return true
  else
    return false
  end
end

# Get user input
```

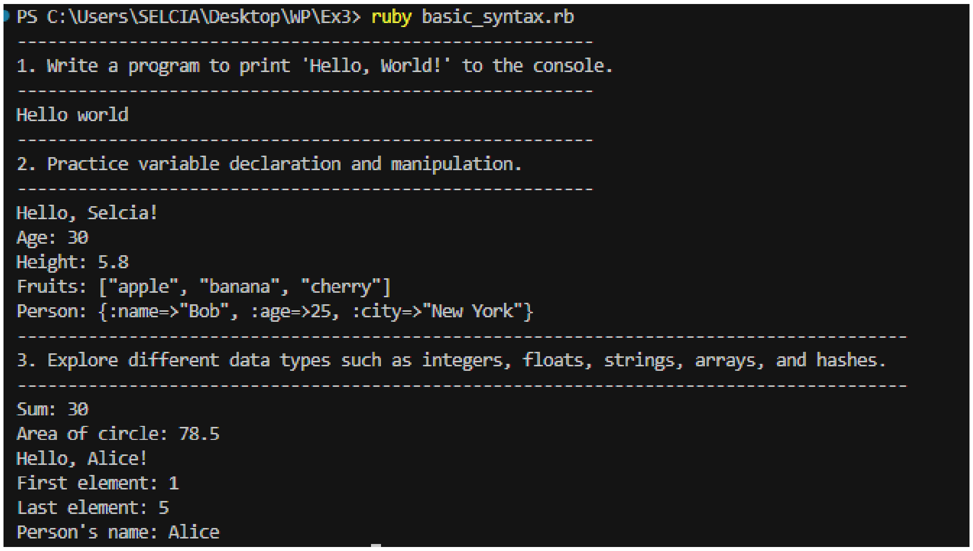
```ruby
puts "Enter the first temperature:"
temp1 = gets.chomp.to_i

puts "Enter the second temperature:"
temp2 = gets.chomp.to_i

# Check temperatures and print the result
result = check_temperatures(temp1, temp2)
puts "Result: #{result}"
```

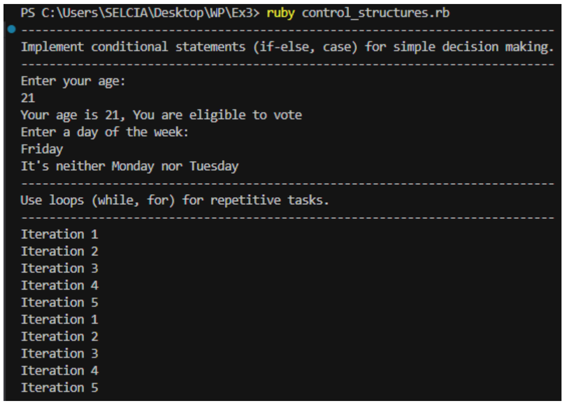
5. Output Screenshots:

1. Basic Syntax Practice:

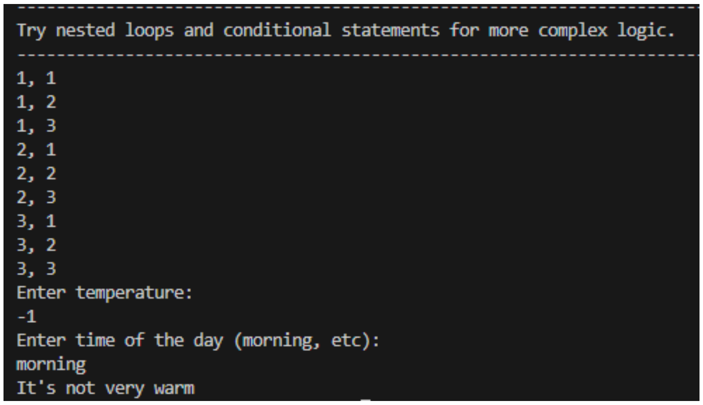


```
PS C:\Users\SELCIA\Desktop\WP\Ex3> ruby basic_syntax.rb
-------------------------------------------------------
1. Write a program to print 'Hello, World!' to the console.
-------------------------------------------------------
Hello world
-------------------------------------------------------
2. Practice variable declaration and manipulation.
-------------------------------------------------------
Hello, Selcia!
Age: 30
Height: 5.8
Fruits: ["apple", "banana", "cherry"]
Person: {:name=>"Bob", :age=>25, :city=>"New York"}
-------------------------------------------------------------------------------------
3. Explore different data types such as integers, floats, strings, arrays, and hashes.
-------------------------------------------------------------------------------------
Sum: 30
Area of circle: 78.5
Hello, Alice!
First element: 1
Last element: 5
Person's name: Alice
```

2. Control Structures:

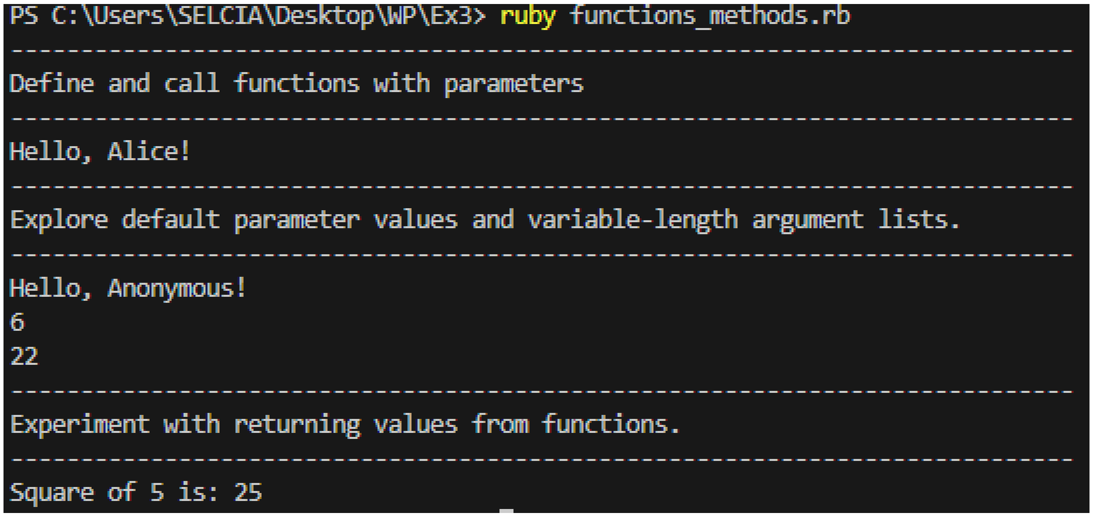


```
PS C:\Users\SELCIA\Desktop\WP\Ex3> ruby control_structures.rb
----------------------------------------------------------------------
Implement conditional statements (if-else, case) for simple decision making.
----------------------------------------------------------------------
Enter your age:
21
Your age is 21, You are eligible to vote
Enter a day of the week:
Friday
It's neither Monday nor Tuesday
----------------------------------------------------------------------
Use loops (while, for) for repetitive tasks.
----------------------------------------------------------------------
Iteration 1
Iteration 2
Iteration 3
Iteration 4
Iteration 5
Iteration 1
Iteration 2
Iteration 3
Iteration 4
Iteration 5
```



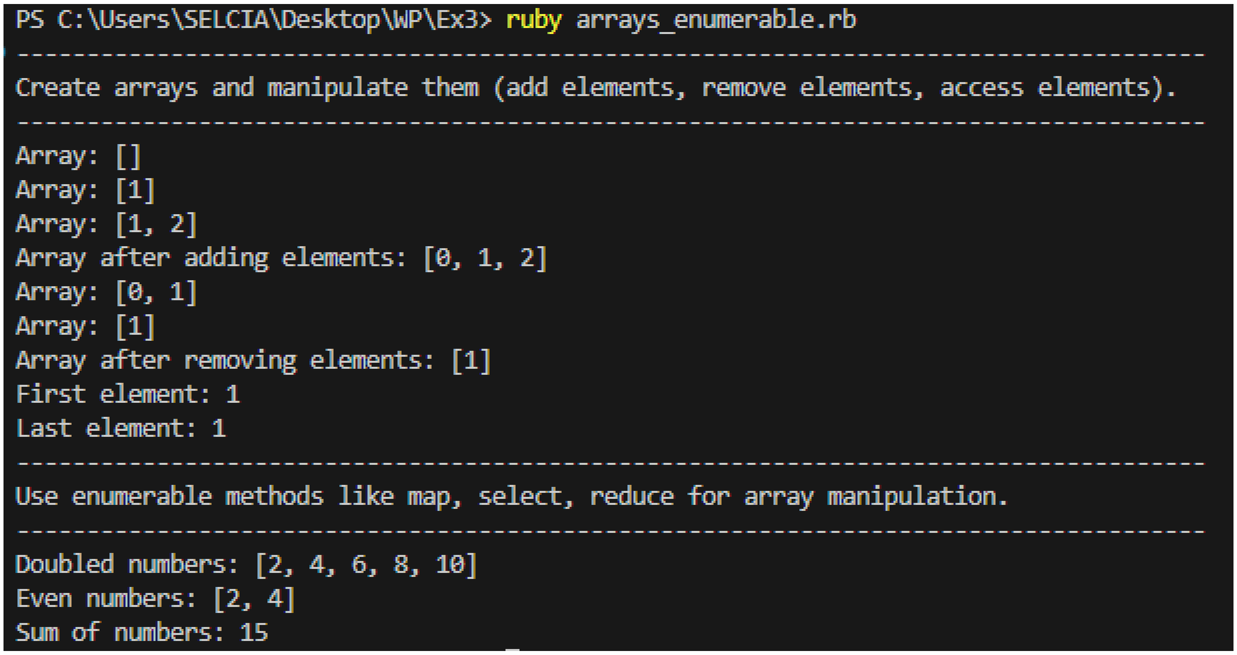
```
-----------------------------------------------------------------------
Try nested loops and conditional statements for more complex logic.
-----------------------------------------------------------------------
1, 1
1, 2
1, 3
2, 1
2, 2
2, 3
3, 1
3, 2
3, 3
Enter temperature:
-1
Enter time of the day (morning, etc):
morning
It's not very warm
```

3. Functions and Methods:



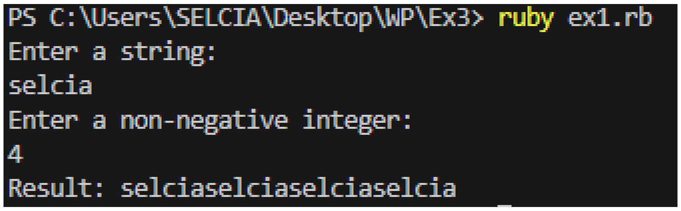
```
PS C:\Users\SELCIA\Desktop\WP\Ex3> ruby functions_methods.rb
----------------------------------------------------------------------------
Define and call functions with parameters
----------------------------------------------------------------------------
Hello, Alice!
----------------------------------------------------------------------------
Explore default parameter values and variable-length argument lists.
----------------------------------------------------------------------------
Hello, Anonymous!
6
22
----------------------------------------------------------------------------
Experiment with returning values from functions.
----------------------------------------------------------------------------
Square of 5 is: 25
```

4. Arrays and Enumerable:



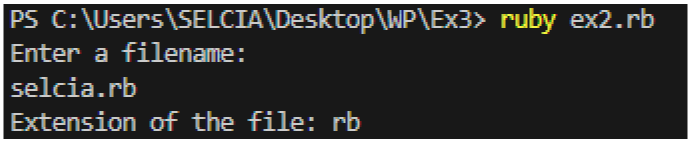
```
PS C:\Users\SELCIA\Desktop\WP\Ex3> ruby arrays_enumerable.rb
------------------------------------------------------------------------------------
Create arrays and manipulate them (add elements, remove elements, access elements).
------------------------------------------------------------------------------------
Array: []
Array: [1]
Array: [1, 2]
Array after adding elements: [0, 1, 2]
Array: [0, 1]
Array: [1]
Array after removing elements: [1]
First element: 1
Last element: 1
------------------------------------------------------------------------------------
Use enumerable methods like map, select, reduce for array manipulation.
------------------------------------------------------------------------------------
Doubled numbers: [2, 4, 6, 8, 10]
Even numbers: [2, 4]
Sum of numbers: 15
```

5. N copies of name:



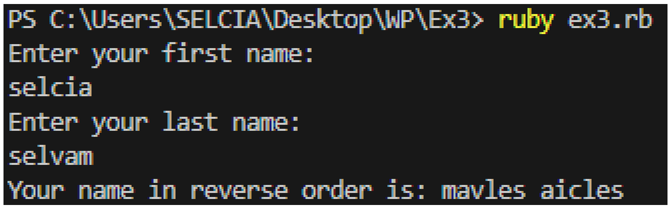
```
PS C:\Users\SELCIA\Desktop\WP\Ex3> ruby ex1.rb
Enter a string:
selcia
Enter a non-negative integer:
4
Result: selciaselciaselciaselcia
```

6. File Extension Script:



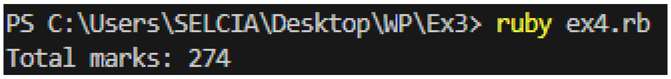
```
PS C:\Users\SELCIA\Desktop\WP\Ex3> ruby ex2.rb
Enter a filename:
selcia.rb
Extension of the file: rb
```

7. Reverse Full Name Script:



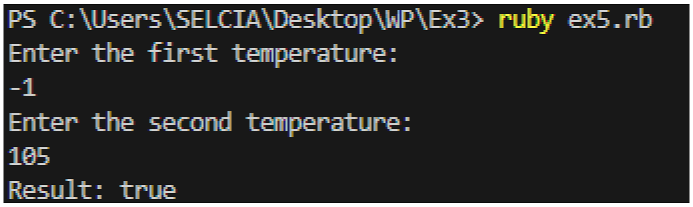
```
PS C:\Users\SELCIA\Desktop\WP\Ex3> ruby ex3.rb
Enter your first name:
selcia
Enter your last name:
selvam
Your name in reverse order is: mavles aicles
```

8. Total Marks Script:



```
PS C:\Users\SELCIA\Desktop\WP\Ex3> ruby ex4.rb
Total marks: 274
```

9. Temperature Check Script:



```
PS C:\Users\SELCIA\Desktop\WP\Ex3> ruby ex5.rb
Enter the first temperature:
-1
Enter the second temperature:
105
Result: true
```

6. Conclusion:

Ruby programming language is practiced by learning various syntax rules and functions.