**Assignment 8**

**Assignment Title:** Ruby Program for Reversing First and Last Name

**Objectives:**

1. To understand basic Ruby environment setup.
2. To write a Ruby program that accepts a user's first and last name.
3. To reverse the order of the first and last name and print them with a space between them.

**Problem Statement:** You are required to write a Ruby program that takes input from the user in the form of their first and last name, and then prints these names in reverse order with a space between them.

**Outcomes:** Upon completion of this assignment, you should be able to:

* Set up a Ruby environment on your system.
* Write a Ruby program that accepts user input.
* Manipulate strings in Ruby to reverse their order.
* Print the manipulated strings.

**Software Specification:**

* Ruby: You will need to have Ruby installed on your system. You can download and install Ruby from the official Ruby website (https://www.ruby-lang.org/en/downloads/).

**Detailed Theory:** Ruby is a dynamic, reflective, object-oriented, and general-purpose programming language. It is known for its simplicity and productivity. Before writing a Ruby program, you need to ensure that you have Ruby installed on your system. Once installed, you can create and execute Ruby programs using a text editor and the command line.

**Additional Theory:**

1. **Ruby Environment Setup:** Setting up a Ruby environment involves installing Ruby on your system. Ruby can be installed on various operating systems such as Windows, macOS, and Linux. Once installed, you can use the command line interface (CLI) or an integrated development environment (IDE) to write and execute Ruby programs.

**Input/Output in Ruby:** Ruby provides several methods for handling user input and output. The **gets** method is commonly used to accept user input from the console, while **puts** and **print** are used to display output. Input obtained through **gets** includes the newline character, which is typically removed using the **chomp** method.

1. **String Manipulation in Ruby:** Ruby offers powerful string manipulation capabilities. Basic string manipulation methods include **concatenation (+)**, **interpolation (#{})**, **length (length)**, **uppercase (upcase)**, **lowercase (downcase)**, **reversal (reverse)**, and more. These methods enable you to manipulate strings according to your requirements, such as reversing their order as done in this assignment.
2. **Concatenation and Interpolation:** Concatenation is the process of combining two or more strings into a single string. In Ruby, concatenation can be achieved using the **+** operator or by using the interpolation syntax **#{}**. Interpolation allows you to embed expressions within strings, making it easier to include variables and expressions in your output.

**Command Line Interface (CLI):** The command line interface is a text-based interface used to interact with the operating system and execute commands. In Ruby programming, the CLI is often used to run Ruby scripts and view their output. It provides a simple and efficient way to write and test Ruby code.

**Algorithm:**

1. Accept the user's input for their first name and store it in a variable.
2. Accept the user's input for their last name and store it in a variable.
3. Concatenate the last name and the first name with a space in between to form the reversed name.
4. Print the reversed name.

**Conclusion**

this assignment provided an opportunity to explore the fundamentals of Ruby programming, focusing on user input, string manipulation, and basic output. By setting up a Ruby environment and writing a simple program, you gained practical experience in interacting with user input, manipulating strings, and producing meaningful output.

Assignment 9

**Title:** Ruby Script to Send an Email

**Objectives:**

* To create a Ruby script that sends an email to a specific user.
* To guide the user through the process of setting up app password for Gmail.
* To demonstrate the usage of the **mail** gem for sending emails in Ruby.

**Problem Statement:** You need to send an email to a specific user using Ruby. Additionally, you need to set up an app password for Gmail to allow the script to send emails securely.

**Outcomes:**

* A Ruby script capable of sending emails using Gmail.
* Knowledge of how to set up app passwords for Gmail.
* Understanding of using the ‘**mail’** gem for email functionality in Ruby.

**Software Specification:**

* Language: Ruby
* External Gem: ‘**mail’**

**Detailed Theory:**

1. **Sending Emails in Ruby:** Sending emails programmatically in Ruby can be achieved using various libraries or gems. One popular gem for this purpose is the **mail** gem. This gem provides a convenient interface for creating and sending emails in Ruby applications. It supports various features such as attachments, HTML content, and SMTP authentication.
2. **The mail Gem:** The **mail** gem is a powerful tool for working with emails in Ruby. It allows you to create email objects, set headers, add attachments, and send emails using SMTP servers. The gem abstracts away the complexities of the SMTP protocol, making it easy to integrate email functionality into your Ruby applications.
3. **Setting up App Passwords for Gmail:** Gmail, like many other email providers, offers two-factor authentication (2FA) for added security. When 2FA is enabled, traditional email clients or scripts may not be able to authenticate using the regular password. In such cases, app passwords are used. An app password is a 16-digit code that allows third-party apps to access your Google Account securely.
4. **Steps to Generate App Password for Gmail:**
   * Go to your Google Account settings.
   * Navigate to the Security section.
   * Enable two-step verification if not already enabled.
   * Look for the option to generate app passwords.
   * Choose the app or device for which you're generating the password (in this case, select "Other" or "Custom").
   * Follow the prompts to generate the app password.
   * Copy the generated app password and use it in your Ruby script for authentication when sending emails via Gmail.
5. **Integration with Replit:** Replit is an online IDE that allows you to write, run, and share code in various programming languages, including Ruby. To integrate the Ruby script for sending emails into Replit:
   * Log in to your Replit account.
   * Create a new Ruby project.
   * Paste the Ruby script into the project.
   * Ensure that the **mail** gem is installed by running **gem install mail** in the Replit shell.
   * Run the Ruby script within Replit to send the email.

**Conclusion :** by following the provided steps, you can create a Ruby script to send emails using Gmail. Setting up an app password ensures secure access to your Gmail account for the script. Utilizing the **mail** gem simplifies the process of sending emails programmatically in Ruby.