

JAVA PROGRAMMING

Tutorial 01

Getting Started

In this tutorial, we will be revisiting the core principles of Pseudo-code and Flowchart. This is an opportunity to apply theoretical knowledge in a practical context, reinforcing your understanding of how to structure and visualize your code before diving into actual programming. We'll begin by translating simple problems into pseudo-code, ensuring a clear, step-by-step approach. Then, we'll create corresponding flowcharts (You can use online tools like *Lucidchart* or *draw.io*) to visually map out the logic.

As we move on to writing basic Java programs, remember that following coding conventions is key. Each exercise should be neatly organized in a separate folder named after its focus; an example shown in Figure 01. This not only helps in maintaining clean code but also in managing your projects efficiently.

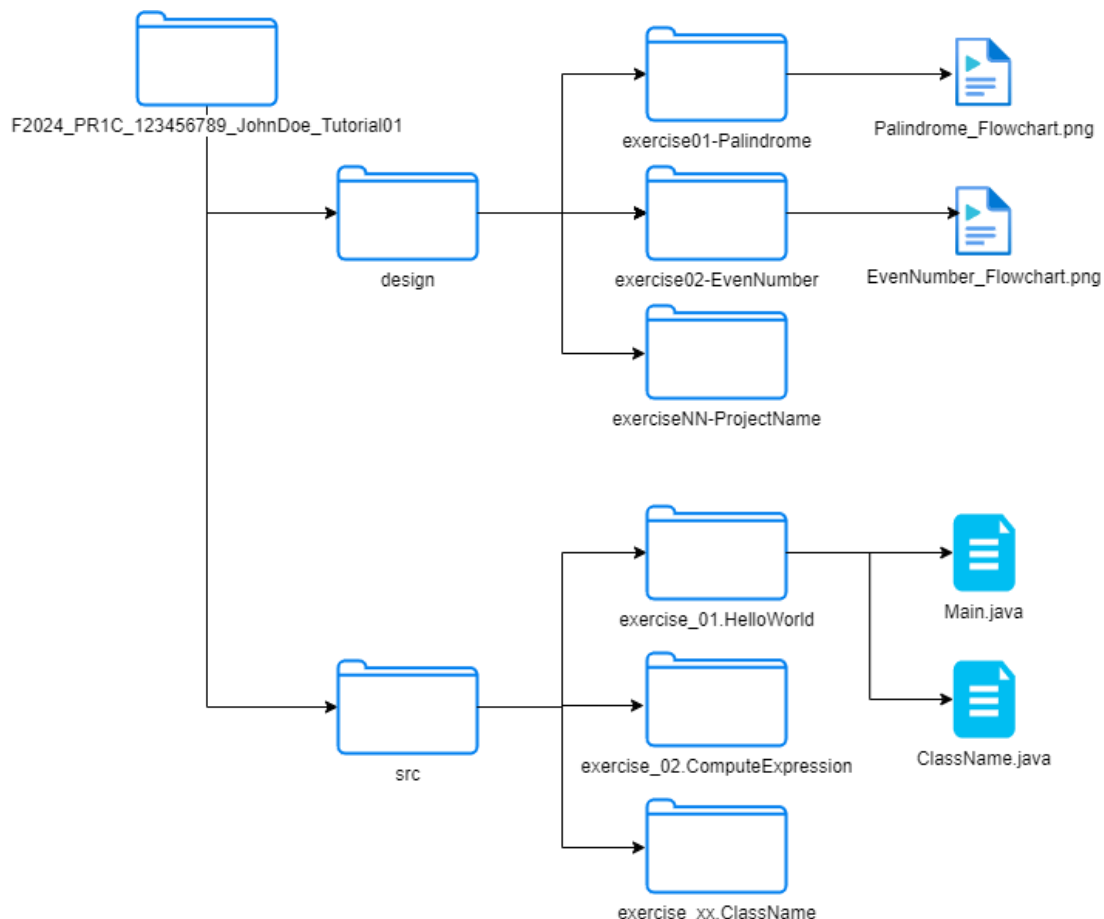


Figure 1. Folders & Naming Guidelines

Finally, we encourage you to push all your materials, including design files and source code, to your Git repository. The guideline shown in Figure 02. This practice not only helps in version control but also prepares you for collaborative work in future projects. Let's start by applying these best practices to create robust, well-documented code from the ground up.

Commit Changes: After completing each exercise, commit the changes to your local Git repository with meaningful commit messages.

Push to GitHub: Regularly push your commits to your GitHub repository.

- **Example:**

bash

Copy code

```
git add .  
git commit -m "Add Exercise01: Print Hello World"  
git push origin main
```

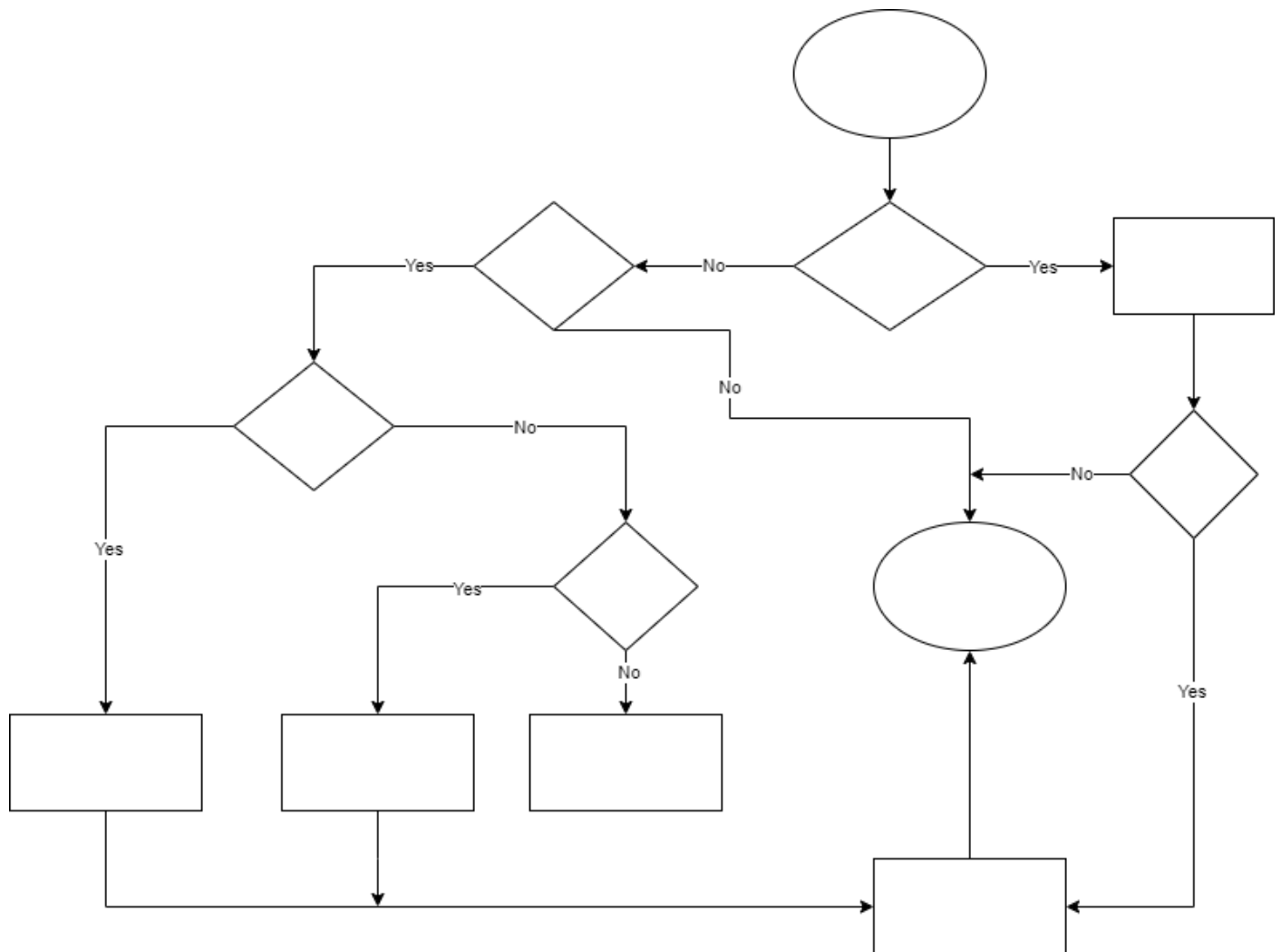
Figure 2. Push changes to remote repositories

In sum, this guide helps students start with good practices in coding and version control while reinforcing the key concepts of the tutorial.

Let's start our journey to the world of programming, and good luck!

Activity 1: Signup Simple Flowchart

Let's warm up by adding the following text to the flowchart (provided on LMS).



1. Start
2. End
3. Already a member?
4. Enter email and password
5. Valid user credentials?
6. Want to login?
7. Login via FB?
8. Fill registration form
9. FB authentication
10. Login via GG?
11. GG authentication
12. User logged in successfully

Activity 2: Even Number Checker

Create a flowchart that checks whether a given number is even or odd. The flowchart should start with an input step where the user provides a number. Then, include a decision-making step to check if the number is divisible by 2, followed by a branch to display the result ("Even" or "Odd").

Activity 3: Temperature Converter

Create a flowchart that visually represents the process of converting temperatures between Celsius and Fahrenheit. The flowchart should begin with a start node and proceed to ask the user to choose whether they want to convert from Celsius to Fahrenheit or vice versa. Based on the user's choice, the flowchart should branch into two paths: one for converting Celsius to Fahrenheit and the other for converting Fahrenheit to Celsius. Each path should include steps for taking the input temperature, performing the necessary conversion calculation, and then displaying the result.

Activity 4: Guessing Game Flowchart

Convert the following Pseudo-code to Flowchart:

```
BEGIN
  DISPLAY "Welcome to the Guessing Game!"

  SET secretNumber TO RANDOM NUMBER BETWEEN 1 AND 100
  SET guess TO 0
  SET attempts TO 0

  WHILE guess IS NOT EQUAL TO secretNumber
    DISPLAY "Enter your guess (between 1 and 100):"
    INPUT guess
    INCREMENT attempts BY 1

    IF guess < secretNumber THEN
      DISPLAY "Too low! Try again."
    ELSE IF guess > secretNumber THEN
      DISPLAY "Too high! Try again."
    ELSE
      DISPLAY "Congratulations! You've guessed the right number."
      DISPLAY "It took you " + attempts + " attempts."
    END IF
  END WHILE

  DISPLAY "Thank you for playing!"
END
```

Activity 5: Hello, World!

Write a program that displays **Hello, World!**.

Activity 6: Compute expressions

Write a program that displays the result of

$$\frac{9.5 \times 4.5 - 2.5 \times 3}{45.5 - 3.5}$$

Activity 7: Population projection

The U.S. Census Bureau projects population based on the following assumptions:

- + One birth every 7 seconds
- + One death every 13 seconds
- + One new immigrant every 45 seconds

Write a program to display the population for each of the next five years. Assume the current population is 312,032,486 and one year has 365 days. *Hint:* In Java, if two integers perform division, the result is an integer. The fractional part is truncated. For example, **5 / 4** is **1** (not **1.25**) and **10 / 4** is **2** (not **2.5**). To get an accurate result with the fractional part, one of the values involved in the division must be a number with a decimal point. For example, **5.0 / 4** is **1.25** and **10 / 4.0** is **2.5**

Note: You do not need to declare any variables!

Activity 8: Algebra

You can use Cramer's rule to solve the following 2 * 2 system of linear equation:

$$\begin{array}{l} ax + by = e \\ cx + dy = f \end{array} \quad x = \frac{ed - bf}{ad - bc} \quad y = \frac{af - ec}{ad - bc}$$

Write a program that solves the following equation and displays the value for x and y:

$$\begin{array}{l} 3.4x + 50.2y = 44.5 \\ 2.1x + .55y = 5.9 \end{array}$$

Submission

For this tutorial, you are required to submit all your Java programs as a single .zip file. Please follow the guidelines below to ensure your submission is well-organized and easy to review:

1. Package Structure:

- ✚ Create a separate package for each exercise. For example, if you have exercises named "**Palindrome**" and "**Pyramid Pattern**", create two packages named *palindrome* and *pyramid_pattern*.
- ✚ Place all the Java packages, files related to each exercise inside the corresponding package.

2. Zipping Your Work:

- ✚ Once you have organized your code into packages, compress the entire project folder into a .zip file.
- ✚ The name of the zip file should follow the format: F2024_PR1C_StudentID_StudentName_TutorialWeekNumber.zip (e.g., F2024_PR1C_123456789_JohnDoe_TutorialWeek01.zip).

3. Submission:

Upload the .zip file to this tutorial's submission box on the course website via the FIT Portal.