SKILL DEVELOPMENT

Project Report

DLithe Consultancy Services Pvt. Ltd.

DLithe™

Be Agile..Be in Demand

Project Report Assessment

Student Name: Vaishnavi Kole

Reg. no: 2JR23CS120

Assignment: Java

Organization: DLithe Consultancy Services Pvt. Ltd.

Supervisor's Name: Archana SM

Submitted to

Signature of Training Supervisor	Signature of Students
Date:	Date:

TABLE OF CONTENTS

INTRODUCTION

In the world of computer programming, solving simple yet logical problems plays a crucial role in building a strong foundation. One such classic problem is checking whether a given input is a palindrome. A palindrome is a sequence of characters that remains the same when read forwards or backwards. This concept applies to words like *madam*, *level*, or numbers like *121* and *1331*. Palindromes are not only interesting from a linguistic point of view, but they also provide a great opportunity to practice basic programming logic.

This project aims to create a Palindrome Checker using the Java programming language. The program will allow users to input a string or number, and it will determine whether the input is a palindrome. Java offers various built-in methods and features that make it easier to handle strings, loops, and condition checks, making it an ideal language for such beginner-friendly projects. This project not only improves coding skills but also strengthens logical thinking and problem-solving abilities.

Overall, the Palindrome Checker is a simple yet effective way to learn how to manipulate strings and use control structures in Java. It can be further enhanced by adding features such as ignoring spaces, punctuation, or even making it case-insensitive for real-world applications. This project is perfect for students and beginners who want to apply their basic Java knowledge to a practical and fun problem.

USE CASE REQUIREMENTS

Use Case

The main use case of this Palindrome Checker application is to allow users to verify whether a given input (word, phrase, or number) is a palindrome. It is especially useful for:

- Students learning Java programming and string manipulation.
- Developers who want to understand logic-based condition checking.
- Interview preparation as it's a common problem in coding assessments.
- Educational tools that teach basic concepts of loops, conditions, and string reversal.

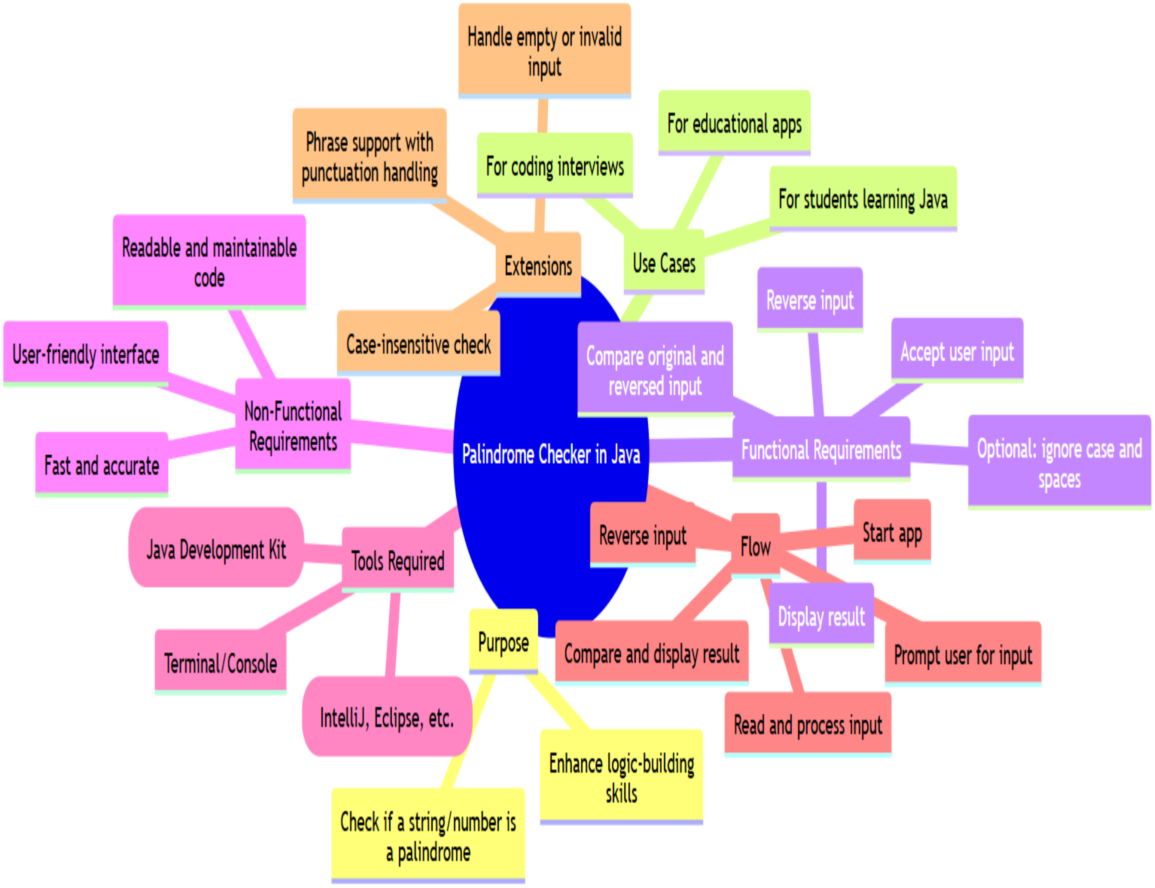
Functional Requirements

1. The program should accept user input (string or number).
2. It should compare the input with its reverse to determine if it's a palindrome.
3. It should display a message indicating whether the input is a palindrome or not.
4. Optionally, it can ignore case sensitivity and spaces for phrases like "A man a plan a canal Panama".

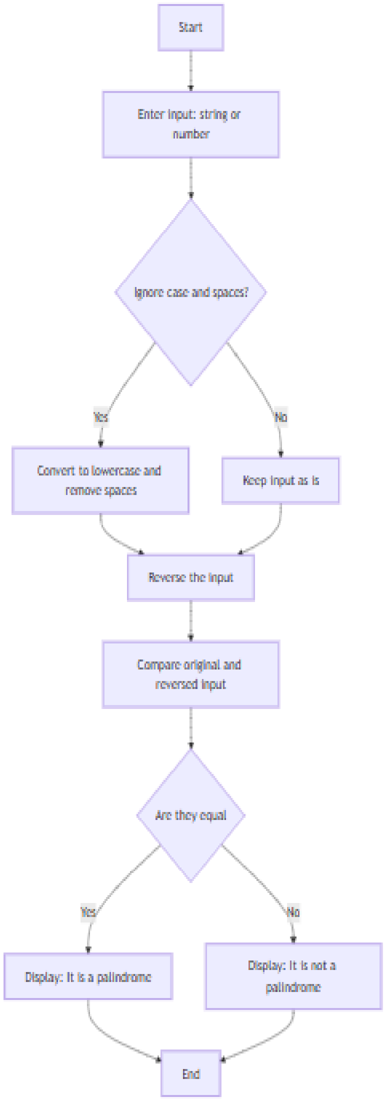
Non-Functional Requirements

1. The program should be simple and user-friendly.
2. It should provide accurate and fast results.
3. The code should be readable and well-documented for learning purposes.

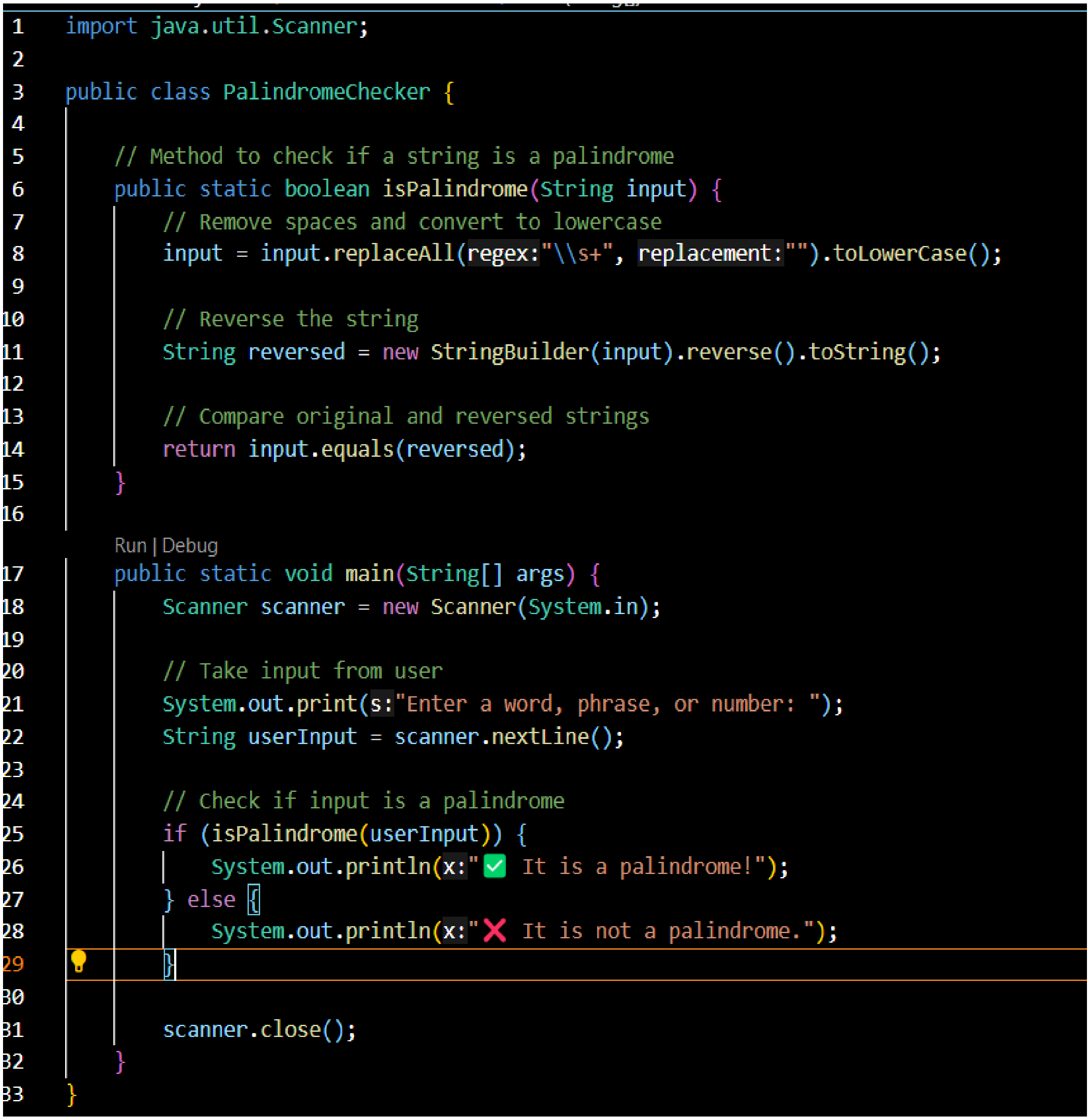
MIND MAP


Palindrome Checker in Java
Extensions
Handle empty or invalid input
Phrase support with punctuation handling
Case-insensitive check
Use Cases
For coding interviews
For educational apps
For students learning Java
Functional Requirements
Reverse input
Accept user input
Compare original and reversed input
Optional: ignore case and spaces
Display result
Flow
Reverse input
Start app
Compare and display result
Prompt user for input
Read and process input
Purpose
Enhance logic-building skills
Check if a string/number is a palindrome
Tools Required
Java Development Kit
Terminal/Console
IntelliJ, Eclipse, etc.
Non-Functional Requirements
Readable and maintainable code
User-friendly interface
Fast and accurate

FLOW CHART

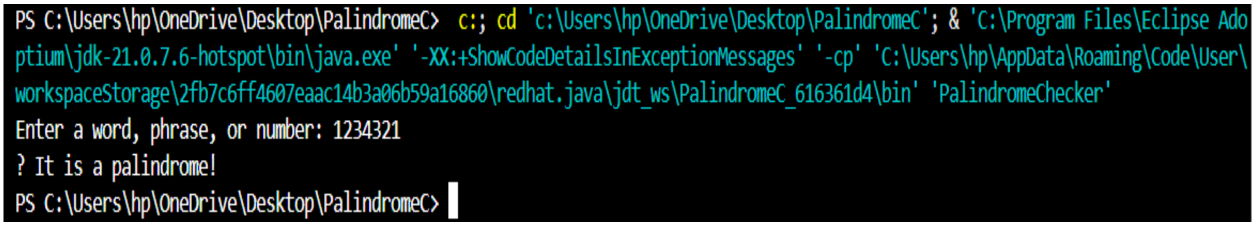

Start
Enter input: string or number
Ignore case and spaces?
Yes
No
Convert to lowercase and remove spaces
Keep input as is
Reverse the input
Compare original and reversed input
Are they equal
Yes
No
Display: It is a palindrome
Display: It is not a palindrome
End

SOURCE CODE

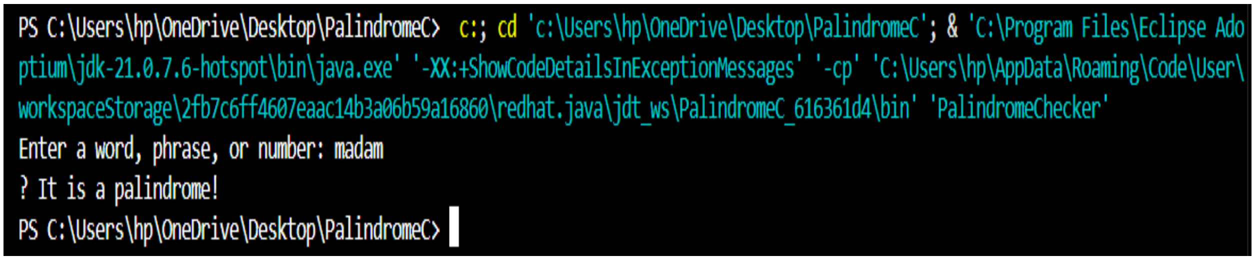


```java
import java.util.Scanner;

public class PalindromeChecker {

    // Method to check if a string is a palindrome
    public static boolean isPalindrome(String input) {
        // Remove spaces and convert to lowercase
        input = input.replaceAll(regex:"\\s+", replacement:"").toLowerCase();

        // Reverse the string
        String reversed = new StringBuilder(input).reverse().toString();

        // Compare original and reversed strings
        return input.equals(reversed);
    }

    Run | Debug
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        // Take input from user
        System.out.print(s:"Enter a word, phrase, or number: ");
        String userInput = scanner.nextLine();

        // Check if input is a palindrome
        if (isPalindrome(userInput)) {
            System.out.println(x:"✅ It is a palindrome!");
        } else {
            System.out.println(x:"❌ It is not a palindrome.");
        }

        scanner.close();
    }
}
```

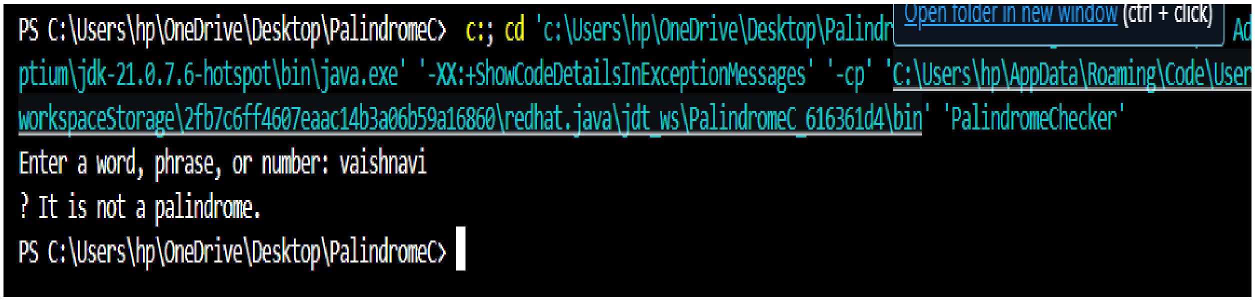
OUTPUT



```
PS C:\Users\hp\OneDrive\Desktop\PalindromeC>  c:; cd 'c:\Users\hp\OneDrive\Desktop\PalindromeC'; & 'C:\Program Files\Eclipse Ado
ptium\jdk-21.0.7.6-hotspot\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\hp\AppData\Roaming\Code\User\
workspaceStorage\2fb7c6ff4607eaac14b3a06b59a16860\redhat.java\jdt_ws\PalindromeC_616361d4\bin' 'PalindromeChecker'
Enter a word, phrase, or number: 1234321
? It is a palindrome!
PS C:\Users\hp\OneDrive\Desktop\PalindromeC>
```



```
PS C:\Users\hp\OneDrive\Desktop\PalindromeC>  c:; cd 'c:\Users\hp\OneDrive\Desktop\PalindromeC'; & 'C:\Program Files\Eclipse Ado
ptium\jdk-21.0.7.6-hotspot\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\hp\AppData\Roaming\Code\User\
workspaceStorage\2fb7c6ff4607eaac14b3a06b59a16860\redhat.java\jdt_ws\PalindromeC_616361d4\bin' 'PalindromeChecker'
Enter a word, phrase, or number: madam
? It is a palindrome!
PS C:\Users\hp\OneDrive\Desktop\PalindromeC>
```



```
PS C:\Users\hp\OneDrive\Desktop\PalindromeC>  c:; cd 'c:\Users\hp\OneDrive\Desktop\Palindr Open folder in new window (ctrl + click)  Ad
ptium\jdk-21.0.7.6-hotspot\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\hp\AppData\Roaming\Code\User
workspaceStorage\2fb7c6ff4607eaac14b3a06b59a16860\redhat.java\jdt_ws\PalindromeC_616361d4\bin' 'PalindromeChecker'
Enter a word, phrase, or number: vaishnavi
? It is not a palindrome.
PS C:\Users\hp\OneDrive\Desktop\PalindromeC>
```

TECHNOLOGIES USED

CHALLENGES

1. **Handling Case Sensitivity**
   - Palindromes like “RaceCar” should be recognized despite uppercase/lowercase letters.
   - Solution: Convert input to all lowercase or uppercase before checking.
2. **Ignoring Spaces and Punctuation**
   - Phrases like “A man, a plan, a canal: Panama” include spaces and punctuation that should be ignored.
   - Solution: Remove spaces and special characters before checking.
3. **Dealing with Empty or Invalid Input**
   - User might enter an empty string or special characters only.
   - Solution: Add input validation to prompt user again or handle gracefully.
4. **Input Type Flexibility**
   - Users may enter numbers, phrases, or single words, which all need to be handled uniformly.
   - Solution: Treat input as a string and apply the same logic.

5. **Performance with Large Inputs**
   - Very long strings might slow down the program if not handled efficiently.
   - Solution: Use efficient string manipulation like StringBuilder for reversal.

6. **User Experience**
   - Making sure the program clearly instructs the user and provides understandable feedback.
   - Solution: Clear prompts and informative messages.

APPLICATIONS

1. **Data Validation & Formatting:**Palindromes are sometimes used in data validation, especially for symmetrical patterns in serial numbers, codes, or passwords.
2. **Cryptography & Security:**Palindromic sequences can be interesting in cryptographic algorithms where symmetric patterns have specific properties useful for encoding or detecting errors.
3. **Bioinformatics**:DNA sequences often contain palindromic patterns that are important in gene regulation and molecular biology research.
4. **Natural Language Processing (NLP)**:Detecting palindromes can be part of linguistic pattern recognition, word games, or text analysis tools.
5. **Coding Interviews & Learning**:Palindrome checking is a classic coding problem used in programming interviews to test string manipulation and logic skills.
6. **Fun & Educational Tools**:Apps or games that challenge users to find or create palindromes help with language learning and cognitive exercises.

CONCLUSION

The Palindrome Checker project is a simple yet powerful way to strengthen your understanding of string manipulation, conditional logic, and user input handling in Java. By building this program, you learn how to process and analyze text data efficiently while considering real-world challenges like case sensitivity and ignoring spaces. Palindrome detection not only serves as a classic programming exercise but also has practical applications in fields ranging from data validation to bioinformatics.

Overall, this project lays a solid foundation for tackling more complex problems and developing confidence in writing clean, readable, and efficient Java code. With further enhancements, such as handling punctuation or creating a user-friendly interface, this basic tool can evolve into a versatile application used in education, cryptography, and beyond.