Chapter 1

ABSTRACT

The Number Guessing Game is a simple interactive project developed to demonstrate the application of fundamental algorithms concepts such as decision making, looping and random number generation. It provides an engaging way to explore basic algorithm design techniques and their implementation in a real-world scenario using C/C++ or Python.

This game is a console-based game in which the computer randomly selects a number within a predefined range, the player attempts to guess the number. After each guess, the computer provides hints(“Too High”, “Too Low”) until the correct number is guessed.

Chapter 2

INTRODUCTION

About the Project

The Number Guessing Game is a simple, interactive command-line program where the user attempts to guess a randomly generated number within a specific range. This project is implemented in the Ada programming language, chosen for its strong typing, safety features, and suitability for teaching fundamental programming concepts.

The core idea behind this game is to use a straightforward brute-force approach where the player iteratively narrows down the range of possibilities based on feedback. Despite its simplicity, this project serves as a good foundation for understanding the flow of control in a program, especially the use of conditional statements (if-else) and loops (while or for). Additionally, it introduces the use of random number generators, which are widely used in gaming and simulation applications.

The game can be implemented in multiple programming languages such as Python or C++. For this particular project, Python is used due to its simplicity and readability, making it easier to focus on the algorithmic aspects rather than the complexities of syntax. However, the underlying logic is universal and can be replicated in any procedural programming language.

This project not only builds logic and problem-solving skills but also demonstrates how user interaction can be incorporated in console-based applications. It emphasizes the importance of validating input, managing control flow, and structuring a clean and maintainable codebase.

Languages/Technologies Used

- **Programming Language**: Python (can also be implemented in C++)
- **Tools**: Any basic code editor (VS Code, PyCharm, Turbo C++, etc.)
- **OS**:Windows/Linux

Chapter 3

SCOPE OF THE PROJECT

Problem Statement

Design and implement a game where a user must guess a randomly selected number between 1 and 100. The system should provide real-time feedback after each guess (too high, too low, or correct) and track the number of attempts taken to guess the number correct.

Design and implement a number guessing game where a user guesses a number generated by the system using basic algorithmic principles.

Objectives

- To understand basic control structures in Ada (loops, conditionals, input/output).
- To implement random number generation using Ada's built-in libraries.
- To develop an interactive program that enhances user engagement.
- To apply simple algorithm design techniques for game logic.

Chapter 4

IMPLEMENTATION

The core logic of the game follows these steps:

1. Generate a random number in the range of 1 to 100.
2. Prompt the user to guess the number.
3. Read the user's input.
4. Compare the guess to the generated number.
5. Provide feedback:
If the guess is higher, display “Too high!”
If the guess is lower, display “Too low!”
If the guess is correct, display “Correct!” and exit the game.
6. Count the number of attempts.
7. Loop until the user guesses correctly.

Chapter 5

ALGORITHM DESIGN TECHNIQUE

The algorithm follows a Brute Force / Trial-and-Error design technique. It repeatedly prompts the user for input and provides guidance based on the input value until the correct answer is found.

The game primarily uses Brute Force Search or Linear Search since the user guesses incrementally based on hints. The design follows a simple Iterative approach with decision-making constructs (if-else).

Chapter 6

SOURCE CODE

```
import java.util.Scanner;
import java.util.Random;
public class NumberGuessingGame {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        Random random = new Random();
        boolean playAgain = true;
        System.out.println("=== Welcome to the Number Guessing Game! ===");
        while (playAgain) {
            int numberToGuess = random.nextInt(100) + 1;
            int guess = 0;
            int attempts = 0;
            System.out.println("\nI've picked a number between 1 and 100. Try to guess it!");
            while (guess != numberToGuess) {
                System.out.print("Enter your guess: ");
                while (!scanner.hasNextInt()) {
                    System.out.print("Invalid input. Enter a number: ");
                    scanner.next(); // clear invalid input
                }
                guess = scanner.nextInt();
                attempts++;
                if (guess < numberToGuess) {
                    System.out.println("Too low! Try again.");
```

```
            } else if (guess > numberToGuess) {
                System.out.println("Too high! Try again.");
            } else {
                System.out.println("🎉 Congratulations! You guessed the number in " + attempts +
" attempts.");
            }
        }
        System.out.print("Do you want to play again? (yes/no): ");
        scanner.nextLine(); // consume newline
        String response = scanner.nextLine().trim().toLowerCase();
        if (!response.equals("yes") && !response.equals("y")) {
            playAgain = false;
            System.out.println("Thanks for playing! Goodbye.");
        }
    }
    scanner.close();
  }
}
```

Chapter 7

OUTPUT

=== Welcome to the Number Guessing Game! ===

I've picked a number between 1 and 100. Try to guess it!
Enter your guess: 99
Too high! Try again.
Enter your guess: 3
Too low! Try again.
Enter your guess: 25
Too low! Try again.
Enter your guess: 79
Too high! Try again.
Enter your guess: 56
? Congratulations! You guessed the number in 5 attempts.
Do you want to play again? (yes/no): no
Thanks for playing! Goodbye.

The game successfully generates a random number and allows the user to guess it through a friendly, interactive loop. Feedback is given after each guess, and the total number of attempts is displayed once the user guesses correctly. The program compiles and runs without errors using GNAT (GNU Ada Compiler).

Chapter 8

CONCLUSION

This project successfully demonstrates how a simple number guessing game can be implemented in Ada using basic programming constructs. It introduces the concepts of user interaction, loop control, conditionals, and random number generation in Ada. The project can be extended further with features such as difficulty levels, score tracking, or a graphical interface for enhanced learning and interactivity.

Chapter 9

REFERENCES

1. Data Structures and Algorithms, Michael T. Goodrich
2. Geeks for Geeks
3. CLRS (Introduction to Algorithms), Cormen et al.
4. ADA Lab Manual and Lecture Notes
5. Fundamentals of Computer Algorithms by Ellis Horowitz