

GUESS THE NUMBER GAME



A PROJECT REPORT

Submitted by

HARI PRIYA E - (2303811710422050)

in partial fulfillment of requirements for the award of the course

CGB1201 - JAVA PROGRAMMING

In

COMPUTER SCIENCE AND ENGINEERING

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(An Autonomous Institution, affiliated to Anna University Chennai and Approved by AICTE, New Delhi)

SAMAYAPURAM – 621 112.

NOVEMBER- 2024

**K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY
(AUTONOMOUS)**

SAMAYAPURAM – 621 112

BONAFIDE CERTIFICATE

Certified that this project report on “ **GUESS THE NUMBER GAME** ”
is the bonafide work of **HARI PRIYA E (2303811710422050)** who carried out
the project work during the academic year 2024 - 2025 under my supervision.

CGB1201-JAVA PROGRAMMING
Dr.A.DELPHIN CAROLINA RANI, M.E., Ph.D.,
HEAD OF THE DEPARTMENT
PROFESSOR

SIGNATURE

Dr.A.Delphin Carolina Rani, M.E., Ph.D.,

HEAD OF THE DEPARTMENT

PROFESSOR

Department of CSE

K.Ramakrishnan College of Technology
(Autonomous)

Samayapuram–621112.

CGB1201-JAVA PROGRAMMING
Mr. M. SARAVANAN, M.E.,
SUPERVISOR
ASSISTANT PROFESSOR

SIGNATURE

Mr. M. Saravanan, M.E.,

SUPERVISOR

ASSISTANT PROFESSOR

Department of CSE

K.Ramakrishnan College of Technology
(Autonomous)

Samayapuram–621112.

Submitted for the viva-voce examination held on 02.12.2024 .

CGB1201-JAVA PROGRAMMING
Mr. MAHARJAN A, M.E.,
INTERNAL EXAMINER
ASSISTANT PROFESSOR

INTERNAL EXAMINER

CGB1201-JAVA PROGRAMMING
Dr. R. SEETHAMILSELVI, M.E., Ph.D.,
EXTERNAL EXAMINER
PROFESSOR
8138-SCE, TRICHY.

EXTERNAL EXAMINER

DECLARATION

I declare that the project report on “**GUESS THE NUMBER GAME**” is the result of original work done by us and best of our knowledge, similar work has not been submitted to “**ANNA UNIVERSITY CHENNAI**” for the requirement of Degree of **BACHELOR OF ENGINEERING**. This project report is submitted on the partial fulfilment of the requirement of the completion of the course **CGB1201 - JAVA PROGRAMMING**.



Signature

HARI PRIYA E

Place: Samayapuram

Date: 02.12.2024

ACKNOWLEDGEMENT

It is with great pride that I express our gratitude and in-debt to our institution “**K. Ramakrishnan College of Technology (Autonomous)**”, for providing us with the opportunity to do this project.

I am glad to credit honourable chairman **Dr. K. RAMAKRISHNAN, B.E.**, for having provided for the facilities during the course of our study in college.

I would like to express our sincere thanks to our beloved Executive Director **Dr. S. KUPPUSAMY, MBA, Ph.D.**, for forwarding to our project and offering adequate duration in completing our project.

I would like to thank **Dr. N. VASUDEVAN, M.Tech., Ph.D.**, Principal, who gave me the opportunity to frame the project with full satisfaction.

I wholeheartedly thank **Dr. A. DELPHIN CAROLINA RANI, M.E., Ph.D.**, Head of the department, **COMPUTER SCIENCE AND ENGINEERING** for providing her encouragement in pursuing this project.

I express our deep expression and sincere gratitude to our project supervisor **Mr. M. SARAVANAN, M.E.**, Department of **COMPUTER SCIENCE AND ENGINEERING**, for his incalculable suggestions, creativity, assistance and patience which motivated us to carry out this project.

I render our sincere thanks to the Course Coordinator and other staff members for providing valuable information during the course.

I wish to express our special thanks to the officials and Lab Technicians of our departments who rendered their help during the period of the work progress.

VISION OF THE INSTITUTION

To serve the society by offering top-notch technical education on par with global standards.

MISSION OF THE INSTITUTION

- Be a center of excellence for technical education in emerging technologies by exceeding the needs of the industry and society.
- Be an institute with world class research facilities.
- Be an institute nurturing talent and enhancing the competency of students to transform them as all-round personality respecting moral and ethical values.

VISION OF DEPARTMENT

To be a center of eminence in creating competent software professionals with research and innovative skills.

MISSION OF DEPARTMENT

M1: Industry Specific: To nurture students in working with various hardware and software platforms inclined with the best practices of industry.

M2: Research: To prepare students for research-oriented activities.

M3: Society: To empower students with the required skills to solve complex technological problems of society.

PROGRAM EDUCATIONAL OBJECTIVES

PEO 1: Domain Knowledge

To produce graduates who have a strong foundation of knowledge and skills in the field of Computer Science and Engineering.

PEO 2: Employability Skills and Research

To produce graduates who are employable in industries/public sector/research Organizations or Work as an entrepreneur.

PEO 3: Ethics and Values

To develop leadership skills and ethically collaborate with society to tackle real-world challenges.

PROGRAM SPECIFIC OUTCOMES (PSOs)

PSO 1: Domain Knowledge

To analyze, design and develop computing solutions by applying foundational concepts of Computer Science and Engineering.

PSO 2: Quality Software

To apply software engineering principles and practices for developing quality software for scientific and business applications.

PSO 3: Innovation Ideas

To adapt to emerging Information and Communication Technologies (ICT) to innovate ideas and solutions to existing/novel problems.

PROGRAM OUTCOMES (POs)

Engineering students will be able to:

- 1. Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
- 2. Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences
- 3. Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations
- 4. Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and

synthesis of the information to provide valid conclusions

- 5. Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations
- 6. The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice
- 7. Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development
- 8. Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
- 9. Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
- 10. Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
- 11. Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
- 12. Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

ABSTRACT

The Guess the Number Game implemented using Java's AWT (Abstract Window Toolkit) and Swing provides an interactive graphical interface for the user to guess a randomly generated number within a specified range. The program uses the AWT and Swing libraries to create a user-friendly interface, which includes a text field for the user to input their guesses, a label to display feedback, and buttons to submit guesses and restart the game. The game generates a random number within a predetermined range (such as 1 to 100), and the user is prompted to guess the number. The application provides immediate feedback, indicating whether the guess is too low, too high, or correct. Once the user correctly guesses the number, the game congratulates them, and the user has the option to start a new round.

The graphical components are arranged using AWT containers like Frame and Panel, and the functionality is enhanced using Swing components such as JLabel, JTextField, and JButton. Events are handled through ActionListeners, where the guess and button clicks trigger responses, updating the display to guide the user through the game process. This game emphasizes object-oriented design and event-driven programming concepts, demonstrating how to use Java's AWT and Swing for GUI-based applications. The purpose of this project is not only to provide an engaging way for users to interact with the program but also to showcase the use of Java's GUI libraries for building simple yet interactive applications.

ABSTRACT WITH POs AND PSOs MAPPING

CO 5 : BUILD JAVA APPLICATIONS FOR SOLVING REAL-TIME PROBLEMS.

ABSTRACT	POs MAPPED	PSOs MAPPED
<p>The Guess the Number Game implemented using Java's AWT (Abstract Window Toolkit) and Swing provides an interactive graphical interface for the user to guess a randomly generated number within a specified range. The program uses the AWT and Swing libraries to create a user-friendly interface, which includes a text field for the user to input their guesses, a label to display feedback, and buttons to submit guesses and restart the game. The game generates a random number within a predetermined range (such as 1 to 100), and the user is prompted to guess the number. The application provides immediate feedback, indicating whether the guess is too low, too high, or correct. Once the user correctly guesses the number, the game congratulates them, and the user has the option to start a new round. The graphical components are arranged using AWT containers like Frame and Panel, and the functionality is enhanced using Swing components such as JLabel, JTextField, and JButton. Events are handled through ActionListeners, where the guess and button clicks trigger responses, updating the display to guide the user through the game process. This game emphasizes object-oriented design and event-driven programming concepts, demonstrating how to use Java's AWT and Swing for GUI-based applications. The purpose of this project is not only to provide an engaging way for users to interact with the program but also to showcase the use of Java's GUI libraries for building simple yet interactive applications.</p>	<p>PO 1 -3 PO 2 -3 PO 3 -3 PO 4 -3 PO 5 -3 PO 6 -3 PO 7 -3 PO 8 -3 PO 9 -3 PO 10 -3 PO 11-3 PO 12 -3</p>	<p>PSO 1 -3 PSO 2 -3 PSO 3 -3</p>

Note: 1- Low, 2-Medium, 3- High

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	
1	INTRODUCTION	
	1.1 Objective	01
	1.2 Overview	01
	1.3 Java Programming concepts	01
2	PROJECT METHODOLOGY	
	2.1 Proposed Work	02
	2.2 Block Diagram	02
3	MODULE DESCRIPTION	
	3.1 Java Swing GUI Components	03
	3.2 Event Handling	03
	3.3 Logic Control	03
	3.4 Game Logic	04
	3.5 Game State Tracking	04
4	CONCLUSION & FUTURE SCOPE	
	4.1 Conclusion	05
	4.2 Future Scope	05
	APPENDIX A (SOURCE CODE)	06
	APPENDIX B (SCREENSHORT)	10
	REFERENCE	11

CHAPTER 1

INTRODUCTION

1.1 Objective

The goal of this project is to create an engaging and interactive game that familiarizes players with logical problem-solving through guess-based gameplay. It demonstrates core Java programming principles, including the use of GUI components, event handling, and effective game state management. The game also introduces modular programming by dividing tasks into separate, manageable methods, ensuring both clarity and maintainability. The inclusion of difficulty levels further enhances the user experience by providing personalized challenges, making the game accessible to players of all skill levels.

1.2 Overview

The Guess the Number game is a fun and educational program designed to test the player's guessing skills. The game randomly generates a number within a specified range based on the chosen difficulty level, and players submit their guesses through a graphical interface. After each guess, feedback such as "Too High," "Too Low," or "Correct!" is displayed to guide the player. The game tracks the number of attempts and ensures the player does not exceed the allowed maximum. It offers three difficulty levels: Easy (range 1–50, maximum 10 attempts), Medium (range 1–100, maximum 7 attempts), and Hard (range 1–200, maximum 5 attempts). Players can restart the game to try again with a new number or exit the application at any time.

1.3 Java Programming Concepts

The Swing library is utilized for GUI development, providing components like JFrame, JLabel, JTextField, and JButton to create an interactive user interface. Event-driven programming is implemented using action listeners that dynamically respond to user inputs, such as button clicks, enabling seamless interaction. The game logic relies on control structures, including conditional statements and modular methods, to handle guess comparisons and gameplay flow. Randomization is achieved using Math.random(), which generates the number the player must guess, ensuring unpredictability and replay value. To maintain robustness, exception handling with try-catch blocks is employed to manage invalid user inputs gracefully. Finally, the application demonstrates encapsulation by organizing related functionalities into methods, ensuring modularity, readability, and maintainability of the code.

CHAPTER 2

PROJECT METHODOLOGY

2.1 Proposed Work

The system provides several features to enhance the gaming experience. Custom difficulty settings allow players to adjust game parameters such as the range of numbers and the maximum number of attempts, tailoring the challenge to their preferences. Interactive gameplay enables users to input their guesses and receive instant feedback, such as whether their guess is "Too High" or "Too Low," making the game engaging and intuitive. Additionally, a retry option allows players to start a new game after either guessing the correct number or exhausting their attempts, ensuring replayability and continuous enjoyment.

2.2 Block Diagram

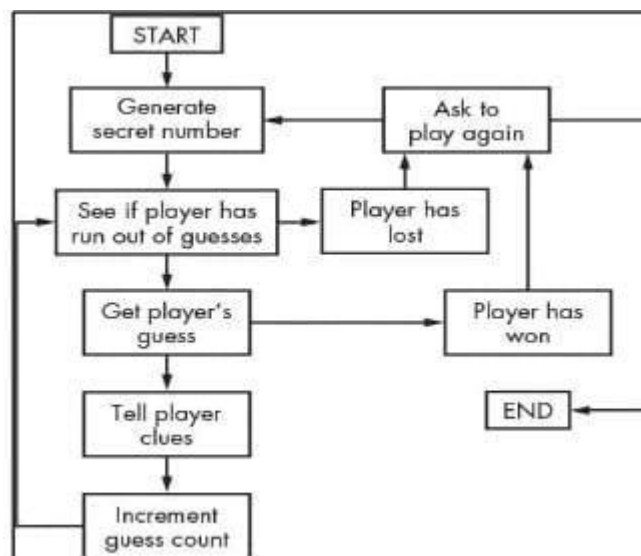


Fig 2.2 Block Diagram

CHAPTER 3

MODULE DESCRIPTION

3.1 Java Swing GUI Components

The graphical interface of the "Guess the Number Game" is built using Java Swing components to provide a user-friendly and interactive experience. JFrame serves as the primary window container for all other elements, forming the backbone of the interface. JLabel is used to display instructions, feedback, and game-related messages to guide the player throughout the gameplay. JTextField allows the player to input their guesses, making it easy for users to interact with the game. JButton is utilized for various actions, including submitting guesses, restarting the game, and exiting the application, ensuring a smooth flow of interactions. Additionally, JOptionPane is employed to present a pop-up dialog for selecting difficulty levels, adding an element of customization to the game's settings. Together, these components provide a seamless and visually appealing user interface.

3.2 Event Handling

The game leverages event handling through ActionListener to respond dynamically to user actions. When a player clicks on a button, the corresponding action listener executes predefined logic, such as processing the entered guess or initiating a new game. This event-driven programming model ensures that the application reacts in real-time, making the gameplay intuitive and engaging. The use of action listeners encapsulates button-specific functionality, enhancing code organization and maintainability.

3.3 Logic Control

Control logic is a fundamental aspect of the game, ensuring it operates smoothly and effectively. Random number generation, implemented using Math.random(), introduces unpredictability by generating a unique number for each game session, which keeps the game fresh and challenging. To enhance robustness, try-catch blocks are used for input validation, preventing the application from crashing due to invalid or non-numeric inputs. This ensures a better user experience by providing meaningful feedback when errors occur, rather than terminating the game unexpectedly.

3.4 Game Logic

The core game logic revolves around comparing the player's guess with the randomly generated number to determine the outcome. Conditional statements (if-else) are used to evaluate whether the guess is too high, too low, or correct, providing immediate feedback to guide the player. The game logic is modularized into dedicated methods for processing guesses and resetting the game, ensuring the code is well-structured and easy to maintain. Modular functions not only improve readability but also make it easier to implement additional features or modify existing ones in the future. This structured approach ensures the game logic is both scalable and adaptable to new requirements.

3.5 Game State Tracking

Game state tracking is essential for managing the player's progress and ensuring smooth gameplay. Key elements include the randomly generated number, the player's attempt count, the maximum allowed attempts, and the difficulty level. These variables are updated dynamically during gameplay to provide real-time feedback and maintain logic flow. Upon game completion, the state adjusts to reflect the outcome, enabling or disabling relevant buttons. The `startNewGame()` method resets all variables for a new session, ensuring no residual data interferes. Effective state management ensures seamless functionality, enhancing the user experience and code maintainability.

CHAPTER 4

CONCLUSION & FUTURE SCOPE

4.1 CONCLUSION

The "Guess the Number Game" demonstrates Java's versatility and its capability to create interactive and dynamic applications. By integrating core programming concepts such as GUI design, event-driven programming, and logical structuring, the project showcases how Java can be used to build engaging and user-friendly applications. The use of Swing components for the interface, randomization for unpredictable gameplay, and exception handling for robust input validation further highlights Java's strengths.

This project serves as an excellent foundation for developers to gain hands-on experience in application development. It not only reinforces fundamental programming skills but also encourages creativity and exploration for future enhancements, such as multiplayer functionality, animations, or mobile adaptations. Ultimately, the game stands as a testament to Java's effectiveness in combining simplicity, flexibility, and functionality, making it a valuable learning experience for aspiring developers.

4.2 FUTURE SCOPE

The game has immense potential for further development and improvement. Future enhancements could include implementing a multiplayer mode, where players compete to guess the number in fewer attempts, or a leaderboard system to track high scores and foster competition. The interface could be improved by incorporating animations and sound effects to make the game more immersive and visually appealing. Additionally, expanding the game to include mobile platforms using JavaFX or other frameworks would make it accessible to a broader audience. Introducing features such as hint systems or customizable difficulty settings could further enrich the gameplay experience. These additions would not only make the game more engaging but also provide developers with valuable insights into advanced programming techniques and real-world software development.

APPENDIX A

(SOURCE CODE)

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class GuessTheNumberGame {
    private static JFrame frame;
    private static JLabel label;
    private static JTextField guessInput;
    private static JButton submitButton;
    private static JButton playAgainButton;
    private static JButton exitButton;
    private static JLabel attemptsLabel;
    private static int randomNumber;
    private static int attempts;
    private static int maxAttempts;
    private static int minRange = 1, maxRange = 100;

    public static void main(String[] args) {
        frame = new JFrame("Guess the Number Game");
        frame.setSize(400, 300);
        frame.setLayout(new FlowLayout());
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        label = new JLabel("Guess the number between " + minRange + " and " + maxRange);
        guessInput = new JTextField(10);
        submitButton = new JButton("Submit Guess");
        playAgainButton = new JButton("Play Again");
        exitButton = new JButton("Exit");
        attemptsLabel = new JLabel("Attempts: 0");
```



```

frame.add(label);
frame.add(guessInput);
frame.add(submitButton);
frame.add(attemptsLabel);
frame.add(playAgainButton);
frame.add(exitButton);

submitButton.setEnabled(true);
playAgainButton.setEnabled(false);
exitButton.setEnabled(true);
submitButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        processGuess();
    }
});

playAgainButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        startNewGame();
    }
});

exitButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        System.exit(0);
    }
});

String[] difficulties = {"Easy", "Medium", "Hard"};
String difficulty = (String) JOptionPane.showInputDialog(frame, "Choose difficulty
level", "Difficulty",
    JOptionPane.QUESTION_MESSAGE, null, difficulties, difficulties[0]);

setDifficulty(difficulty);

```

```

        startNewGame();

        frame.setVisible(true);
    }

    private static void setDifficulty(String difficulty) {
        switch (difficulty) {
            case "Easy":
                maxRange = 50;
                maxAttempts = 10;
                break;
            case "Medium":
                maxRange = 100;
                maxAttempts = 7;
                break;
            case "Hard":
                maxRange = 200;
                maxAttempts = 5;
                break;
        }
        label.setText("Guess the number between " + minRange + " and " + maxRange);
    }

    private static void startNewGame() {
        randomNumber = (int) (Math.random() * (maxRange - minRange + 1)) +
minRange;
        attempts = 0;
        attemptsLabel.setText("Attempts: 0");
        guessInput.setText("");
        submitButton.setEnabled(true);
        playAgainButton.setEnabled(false);
    }

    private static void processGuess() {

```

```

try {
    int guess = Integer.parseInt(guessInput.getText());
    attempts++;
    attemptsLabel.setText("Attempts: " + attempts);

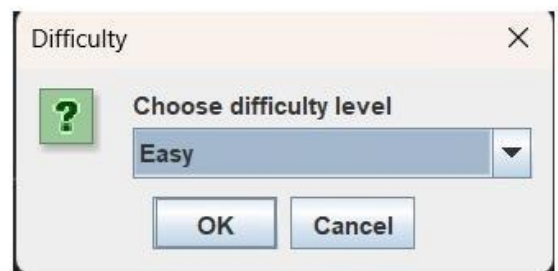
    if (guess < randomNumber) {
        label.setText("Too Low! Try again.");
    } else if (guess > randomNumber) {
        label.setText("Too High! Try again.");
    } else {
        label.setText("Correct! You guessed the number!");
        submitButton.setEnabled(false);
        playAgainButton.setEnabled(true);
    }

    if (attempts >= maxAttempts && guess != randomNumber) {
        label.setText("You've reached the max attempts. The number was " +
randomNumber);
        submitButton.setEnabled(false);
        playAgainButton.setEnabled(true);
    }
} catch (NumberFormatException e) {
    label.setText("Please enter a valid number.");
}
}
}

```

APPENDIX B

OUTPUT



REFERENCES

- Herbert Schildt, Java: The Complete Reference, McGraw Hill Education.
- GeeksforGeeks Java Programming:
<https://www.geeksforgeeks.org/java/Teluska>
- Java Tutorials:
<https://www.youtube.com/user/javaboynavin>