Understanding Python With a Number Guessing Game

Get Ready to Code: An Exciting Introduction to Programming Fun

Presented by:

- **Bikash Das (ADTU/0/2024-28/BCSM/071)**
- **Nayandeep Goswami (ADTU/0/2024-28/BCSM/047)**
- **Suraj Choudhary (ADTU/0/2024-28/BCSM/075)**
- **Asif Raza (ADTU/0/2024-28/BCSM/004)**
- **Dhiman Saikia (ADTU/0/2024-28/BCSM/062)**

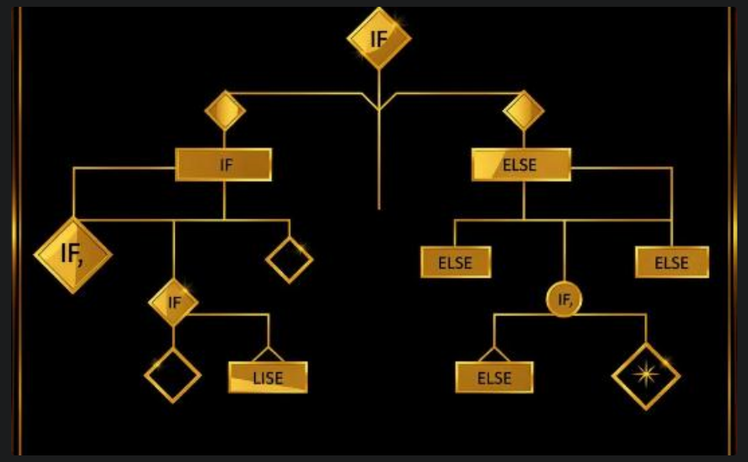
[Presentation Date : 19/05/2025]

Table of Contents

This presentation covers a Python number guessing game.

Introduction

Welcome to the Number Guessing Game! The objective is to guess a randomly generated number between 1 and 20. The game provides hints to help us refine our guesses, making it an engaging and interactive learning experience.



Interactive Fun

Engage in an interactive game that challenges your guessing skills.

Random Number Generation

Experience the excitement of guessing a new random number each time.

Strategic Hints

Receive helpful hints to guide your guesses and improve your strategy.

Objective

Improve Logic

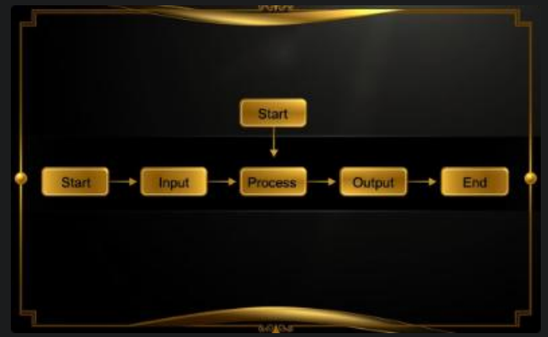
Sharpen problem-solving skills.

Learn Programming Ideas

Understand loops and random choices.

Applying Concepts

Master Python through practice.



Algorithmic Thinking

Follow the program's clear logic.

Tools Used

Python

Core language for game logic and input.

Random Module

Generates the number to guess.

Control Structures

Manages input, validates guesses, and controls game flow.

Exception Handling

Handles invalid inputs to prevent crashes.

User Input Handling

Receives player guesses.

Scoring System

Dynamic scoring based on attempts.

Hint System

Provides clues after incorrect attempts.

Concepts Used

1 Random Module

Generates pseudo-random integers using the **randint()** function.

2 If-Else Statements

Compares guesses to the target number and provides feedback accordingly.

3 While Loop

Continues running the game until the correct guess is made or the player exits, repeatedly prompting for input.

4 OOPs

Uses classes to organize game variables and methods for better structure.

5 Tkinter

Builds the game's graphical user interface.

6 File Handling

Saves player profiles and high scores in JSON format for persistent storage.

Process Flow Diagram

1

User Guess Input

- Players enter a number guess between 1 and 20.

2

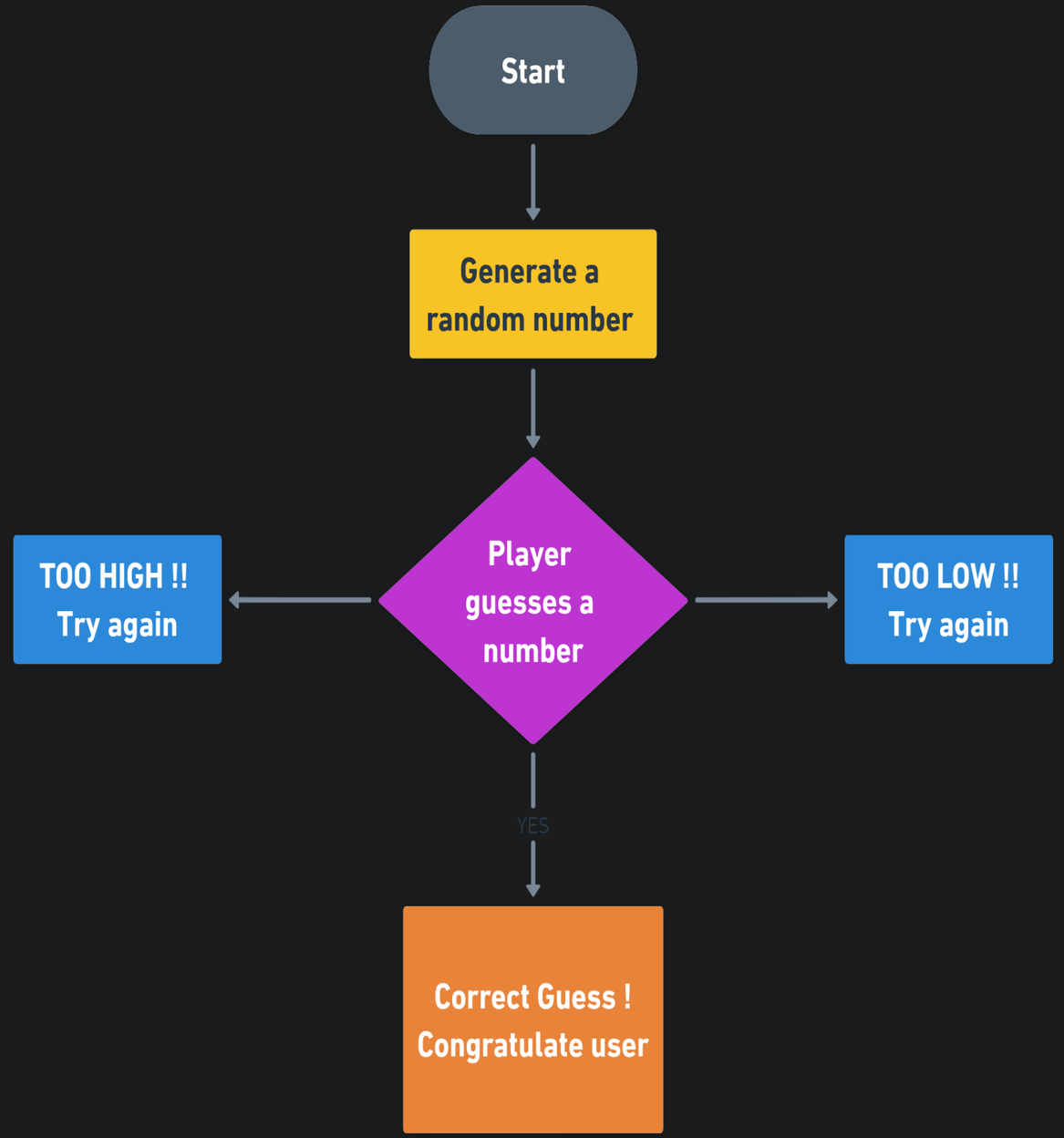
Random Number Generation

- The system randomly selects a number at the start of the game.

3

Hint-Based Feedback

- The game provides hints (Too High / Too Low) after each guess.



Conclusion

1

Beginner-Friendly

Simple structure and basic Python make it easy for new programmers to understand and change the code.

2

Core Concepts

Shows how to create random numbers, use `while` loops, and use `if-else` statements for user feedback.

3

Enhancements

Easily add difficulty levels, scoring, or limit the number of guesses allowed.

4

Fun Learning

Interactive and engaging, it helps learn programming through feedback and encourages understanding of conditional logic.

Thank You