Number Guessing Game (C-Language)

Project for semester 1st

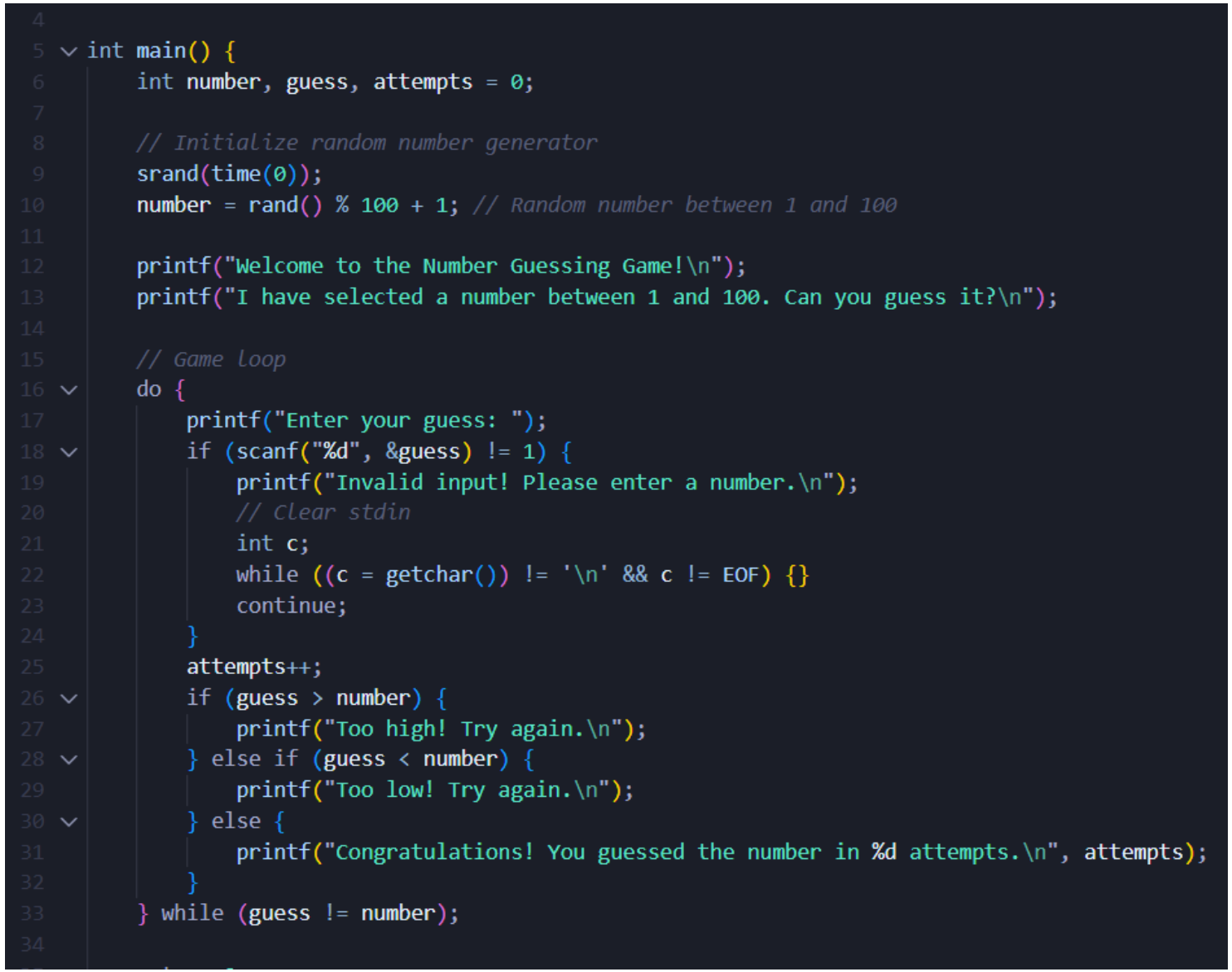
2025-2026

Hardik Soni (PPT for explaining working of code and game)

Code of the Game (Image)

Overview of game:-

This program picks a random number between 1 and 100, then repeatedly asks the user to guess it, giving feedback (“too high”/“too low”) until the correct guess is made. It also handles invalid input and counts the number of attempts.



```c
int main() {
    int number, guess, attempts = 0;

    // Initialize random number generator
    srand(time(0));
    number = rand() % 100 + 1; // Random number between 1 and 100

    printf("Welcome to the Number Guessing Game!\n");
    printf("I have selected a number between 1 and 100. Can you guess it?\n");

    // Game loop
    do {
        printf("Enter your guess: ");
        if (scanf("%d", &guess) != 1) {
            printf("Invalid input! Please enter a number.\n");
            // Clear stdin
            int c;
            while ((c = getchar()) != '\n' && c != EOF) {}
            continue;
        }
        attempts++;
        if (guess > number) {
            printf("Too high! Try again.\n");
        } else if (guess < number) {
            printf("Too low! Try again.\n");
        } else {
            printf("Congratulations! You guessed the number in %d attempts.\n", attempts);
        }
    } while (guess != number);
```

```c
#include <stdio.h>
#include <stdlib.h>
#include <time.h>


int main() {
    int number, guess, attempts = 0;

    // Initialize random number generator
    srand(time(0));
    number = rand() % 100 + 1; // Random number between 1 and 100

    printf("Welcome to the Number Guessing Game!\n");
    printf("I have selected a number between 1 and 100. Can you guess it?\n");

    // Game loop
    do {
        printf("Enter your guess: ");
        if (scanf("%d", &guess) != 1) {
            printf("Invalid input! Please enter a number.\n");
            // Clear stdin
            int c;
            while ((c = getchar()) != '\n' && c != EOF) {}
            continue;
        }
        attempts++;
        if (guess > number) {
            printf("Too high! Try again.\n");
        } else if (guess < number) {
            printf("Too low! Try again.\n");
        } else {
            printf("Congratulations! You guessed the number in %d attempts.\n", attempts);
        }
    } while (guess != number);

    return 0;
}
```

Code of the Game (Text)

Explanation Begin
Step By Step

#include <stdio.h>, #include <stdlib.h>, #include <time.h> (Line- 1_3 in code)

Explanation:-

- <stdio.h> (Standard Input/Output header)

- Provides functions like printf, scanf, getchar, puts.

- Without it, you couldn't print messages or read input from the user.

- <stdlib.h> (Standard Library header)

- Provides general utility functions.

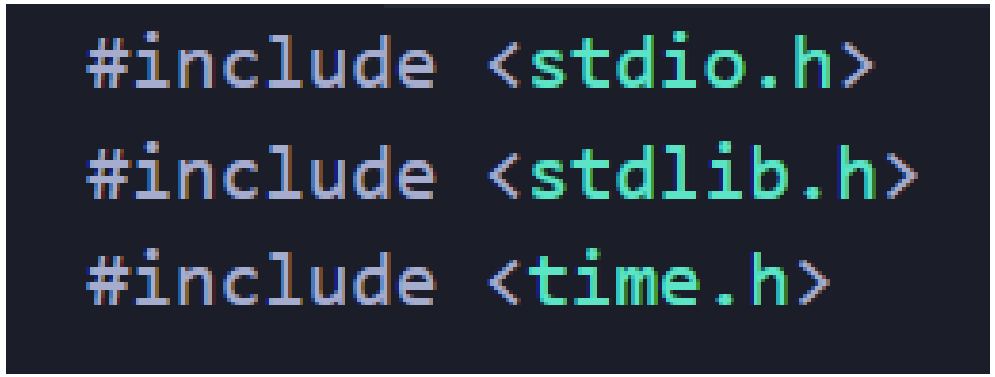
- In your program, it's needed for rand() and srand().

- Also includes functions like malloc, free, exit.

- <time.h> (Time header)

- Provides functions for working with time.

- You used time(0) to seed the random number generator with the current time, ensuring different random numbers each run



int number, guess, attempts = 0; (Line- 5_6 in code)

Explanation:-

- number: Holds the secret random number.

- guess: Stores the user's current guess.

- attempts: Counts how many valid guesses the user has made (starts at 0).

```
int main() {
    int number, guess, attempts = 0;
```

srand(time(0));number = rand() % 100 + 1; (Line- 9_10 in code)

Explanation:-

- srand(time(0)): Seeds the random generator with the current time so each run is different.

- rand() % 100 + 1: Generates an integer in [1, 100]. Modulo 100 gives 0–99; adding 1 shifts to 1–100.

```
srand(time(0));
number = rand() % 100 + 1; // Random number between 1 and 100
```

(Line- 16_33 in code)

Explanation:-

- Prompting for input:

- Label: printf("Enter your guess: ")

- What it does: Asks the user for a number each round.

- Input validation:

- Condition: scanf("%d", &guess) != 1 checks if reading an integer succeeded.

- On failure: Prints an error, then clears the input buffer

```c
// Game loop
do {
    printf("Enter your guess: ");
    if (scanf("%d", &guess) != 1) {
        printf("Invalid input! Please enter a number.\n");
        // Clear stdin
        int c;
        while ((c = getchar()) != '\n' && c != EOF) {}
        continue;
    }
    attempts++;
    if (guess > number) {
        printf("Too high! Try again.\n");
    } else if (guess < number) {
        printf("Too low! Try again.\n");
    } else {
        printf("Congratulations! You guessed the number in %d attempts.\n", attempts);
    }
} while (guess != number);
```

int c;while ((c = getchar()) != '\n' && c != EOF) {} (Line- 21_23 in code)

Explanation:-

- Purpose: Discards leftover characters (like letters) up to the newline so the next scanf won't immediately fail.

- continue; Skips the rest of the loop and asks again without incrementing attempts.

- Counting attempts:

- Line: attempts++;

- Scope: Only increments when a valid integer was entered.

- Comparing the guess:

- Too high: guess > number → prints "Too high!"

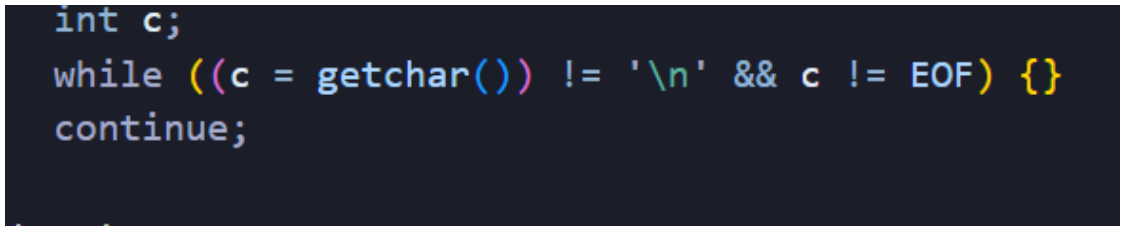
- Too low: guess < number → prints "Too low!"

- Correct: else branch prints the success message with attempts.

- Loop condition:

- Line: } while (guess != number);

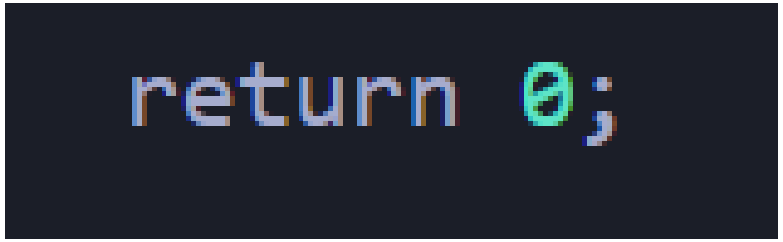
- Effect: The loop continues until the correct guess is made



return 0; (Line- 35 in code)

Explanation:-

- Exit status: Returns 0 to the operating system, indicating successful termination.



Sample Output

What happens in a sample run

Start: Program picks, say, 57.

User enters “abc”: Fails scanf, buffer is cleared, prompt repeats.

User enters 80: Valid; attempts = 1; prints “Too high!”.

User enters 50: attempts = 2; prints “Too low!”.

User enters 57: attempts = 3; prints the congratulations message; loop exits.

Thank You for reading and understanding the code please run it on your own vs-code and understand this perfectly.

Have a nice day

Warm Regards:- Hardik Soni