Week 2-Task 1

PROJECT
PERSONAL INFO

PREPARED BY

Sana Fatima

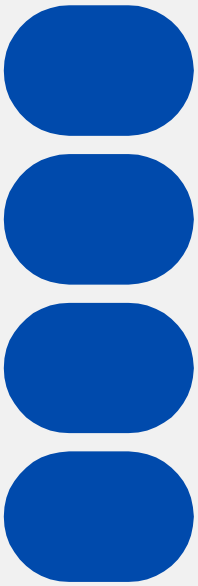
sana.faati@gmail.com

PREPARED TO

A&D Tech

ad.techinnov25@gmail.com

TABLE OF CONTENTS



Problem Statement

Implementing the Program

- Create a new project in your development environment and name it GuessingGame.
- Use your language's random number generation to pick a secret number between -100 and 50.
- Prompt the user to guess the number. Read their input from the console.
- Implement a loop that checks the user’s guess against the secret number.
- If the guess is too high or too low, inform the user and allow them to guess again.

- If the guess is correct, congratulate the user and break out of the loop.
- Keep track of the number of attempts and display this number when the user
- guesses correctly.

SOLUTION OVER VIEW

The solution overview for this program is:

This program is a simple number guessing game where the user attempts to guess a randomly generated number within a specified range. The game provides feedback if the guess is too high or too low and continues until the user guesses the correct number. The program also tracks and displays the number of attempts made by the user to guess the correct number.

Components and Functionality

1.Headers and Namespaces

- **#include <iostream>**: This header is included to facilitate input and output operations.
- **#include <cstdlib>**: This header is included to use the rand() and srand() functions for generating random numbers.
- **#include <ctime>**: This header is included to use the time() function to seed the random number generator.
- **using namespace std;** This line is included to avoid prefixing standard library functions and objects with std::.

2.Main Function

- **Variable Declarations**:
  - int number, guessnum, Variables to store the random number and the user's guess.
  - int upper = 50; int lower = -100;: Constants defining the upper and lower bounds for the random number.
  - int numberofattempts = 0;: Variable to count the number of attempts made by the user.
- **Random Number Generation**:
  - srand(time(0));: Seeds the random number generator with the current time to ensure a different sequence of random numbers each run.
  - number = (rand() % (upper - lower + 1)) + lower;: Generates a random number between lower and upper.

3.Gameplay Loop:

- User Input: cin >> guessnum; takes the user's guess.
- Increment Attempt Counter: numberofattempts++; increments the count of attempts.
- If the guess is higher than the random number, output "Too high, Guess again".

- If the guess is lower than the random number, output "Too low, Guess again". If the guess is correct, output "Congratulations! Correct guess" along with the number of attempts.
- The loop continues until the user guesses the correct number (while (guessnum != number);).

CHALLENGES AND RESOLUTIONS

.

Challenge:
The program initially did not compile due to the missing #include <cstdlib> header, which caused errors related to the rand() and srand() functions not being declared and I do not have knowledge how to generate a random number

Resolution:
Firstly, I have consulted a you-tube video to understanding basically how a random number is generated. Including the #include <cstdlib> header resolved this issue, as this header contains the declarations for these functions. The <ctime> header was also included for the time() function.

INSTRUCTIONS

1. Open an integrated development environment (IDE) and create a new project named guessing_game
2. Write a code.
3. Compile the Code and run.
4. Enter a guess and press Enter.
5. The program will inform you if your guess is too high, too low, or correct.
6. Continue guessing until you guess the correct number.
7. The program will display the number of attempts it took to guess the correct number.
8. The game ends when you correctly guess the number. The program will display a congratulatory message along with the number of attempts you made.