

Assignment at end of session

◆ Assignment 1: Number Guessing Game

Problem Statement:

Create a game where the computer randomly selects a number between 1 and 50. The player has 5 chances to guess the number. After each guess, provide a hint:

- “Too High” if the guess is more
- “Too Low” if the guess is less
- “Correct!” if the guess matches

Instructions for Students:

- Use `import random` for number generation
- Use `input()` to take user guess
- Use `if-elif-else` conditions to compare
- Explain each line of code

Outcome Knowledge:

- Understanding of variables, random number generation
 - Using conditional statements and loops
 - Basic input/output handling
-

◆ Assignment 2: Word Scramble Game

Problem Statement:

Design a game that scrambles a word (e.g., "python" → "nothpy") and asks the user to guess the original word. Give 3 attempts. At the end, display the correct word if the user fails.

Instructions for Students:

- Use list/strings and `random.shuffle()`
- Take user input and compare
- Keep track of number of attempts

Outcome Knowledge:

- String manipulation, lists, random functions
Loops and conditionals
User interaction and logic building

◆ Assignment 3: Rock-Paper-Scissors Game

Problem Statement:

Make a 2-player (user vs computer) Rock-Paper-Scissors game. Each round, both players choose an option, and the winner is decided based on the rules. Display score after 3 rounds.

Instructions for Students:

- Use `input()` for user and `random.choice()` for computer
- Use `if-elif` to determine the winner
- Keep score using variables

Outcome Knowledge:

- Decision-making using conditionals
 - Using loops for multiple rounds
 - Handling user input and logic
-

◆ Assignment 4: Dice Rolling Simulator

Problem Statement:

Simulate the rolling of a dice. Allow the user to roll the dice multiple times by entering 'yes' or 'no'. Print a random number between 1 and 6 for each roll.

Instructions for Students:

- Use `random.randint(1,6)`
- Loop until user decides to stop
- Display each result with a friendly message

Outcome Knowledge:

- Random number generation
 - Loop control with `while` and user input
 - Data display and user interaction
-

◆ Assignment 5: Hangman (Mini Version)

Problem Statement:

Create a mini Hangman game where the user has to guess a 5-letter word. The user can guess one letter at a time, with a maximum of 6 incorrect guesses.

Instructions for Students:

- Predefine a word (like "apple")
- Use loops, `in` operator for checking letters
- Display correct letters in their positions

Outcome Knowledge:

- Working with strings and lists
 - Logic design with conditionals and loops
 - Interactive coding using `input()` and feedback system
-