Fizz-Buzz Game

This document explains the concepts being used in the FizzBuzzGame.py program, the logic behind each implementation, and the purpose of different approaches.

1. Basic Game (Procedural Programming)

This section demonstrates how the FizzBuzz game can be implemented using procedural programming concepts.

* Uses a simple for loop from 1 to 100.
* Checks divisibility of each number by 3, 5, or both.
* Prints 'Fizz' if divisible by 3, 'Buzz' if divisible by 5, and 'FizzBuzz' if divisible by both.
* Otherwise, it prints the number itself.

2. Game using Object-Oriented Programming (OOP)

This section shows how the same FizzBuzz logic can be applied using OOP concepts.

* Encapsulates the logic inside a class called 'FizzBuzzGame'.
* Defines a method 'play()' that runs the FizzBuzz logic.
* An object of the class is created and the method is called to execute the game.

3. Interactive Game (Project Requirement)

This version makes the game interactive by involving user input.

* Uses the 'random' library to generate random numbers between 1 and 100.
* Displays the rules of the game to the user.
* Asks the player to input the correct response (Fizz, Buzz, FizzBuzz, or number).
* If the user’s answer is wrong, the game ends and the correct answer is displayed.
* If correct, the game continues with a new random number added to the previous one.

4. Handling Input Variations

To make the game more user-friendly, input handling is improved using string methods.

* Uses '.strip()' to remove extra spaces from user input.
* Uses '.lower()' to handle case sensitivity, ensuring 'fizz', 'Fizz', or ' FIZZ ' are treated the same.
* Compares the lowercase version of the expected answer with the user's input.