**Name: Sundas Shoukat**

**Section: BSDS-2A**

**Roll no: 057**

**Miniproject 1: Fizz Buzz**

**Requirments:**

1.Print numbers from 1 to 100.  
2. If a number is divisible by 3, print 'Fizz'.  
3. If a number is divisible by 5, print 'Buzz'.  
4. If a number is divisible by both 3 and 5, print 'Fizz Buzz'.  
5. Otherwise, just print the number.

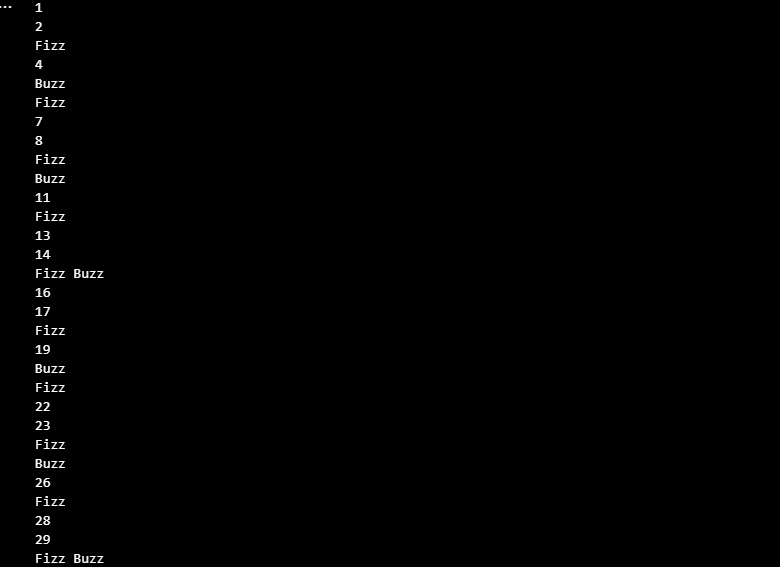
**Explanation of the code:**

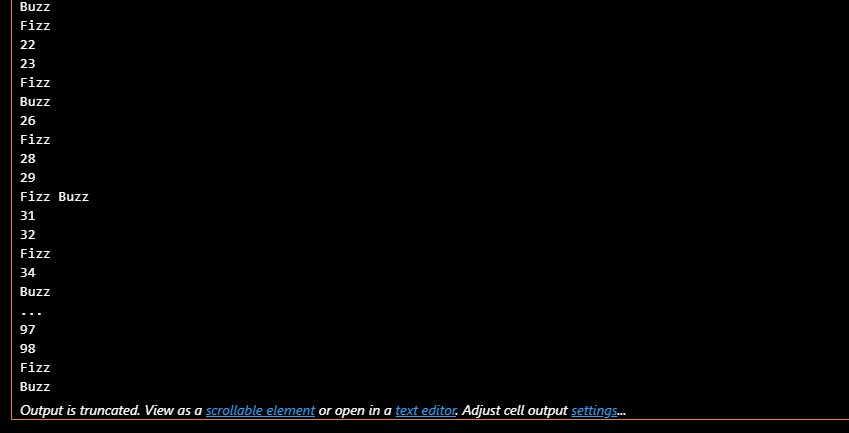
We solved this problem using Object-Oriented Programming (OOP). A class named FizzBuzz was created. The constructor (\_\_init\_\_) initializes the limit, which by default is 100. Inside the class, we created a method called play() that contains the main logic of the game.  
  
The play method uses a loop that goes from 1 to the limit. For each number, the program checks:  
- If divisible by 3 and 5 → print 'Fizz Buzz'  
- If divisible only by 3 → print 'Fizz'  
- If divisible only by 5 → print 'Buzz'  
- Otherwise, print the number itself.

**Why this approach:**

This approach is simple and clear. Using a class makes the program structured and reusable. The modulo (%) operator is the easiest way to check divisibility. This makes the solution short, readable, and easy to understand.

**Output:**

****

****

**Mini Project 2: Movie Budgets**

**Requirements:**

1. Calculate the average budget of all movies.  
2. Print movies that have a budget higher than the average, along with how much higher they are.  
3. Print the total number of movies with a budget higher than the average.  
4. (Optional) Allow the user to add more movies before running calculations.

**Explanation of the Code:**

We created a class called MovieBudget. Inside this class, the constructor (\_\_init\_\_) takes a list of movies, where each movie is a tuple (name, budget).  
  
The program first calculates the average budget by adding all budgets and dividing by the total number of movies.  
  
Next, it loops through the list of movies. If a movie’s budget is higher than the average, it prints the movie name and shows how much higher the budget is. It also counts how many such movies exist.

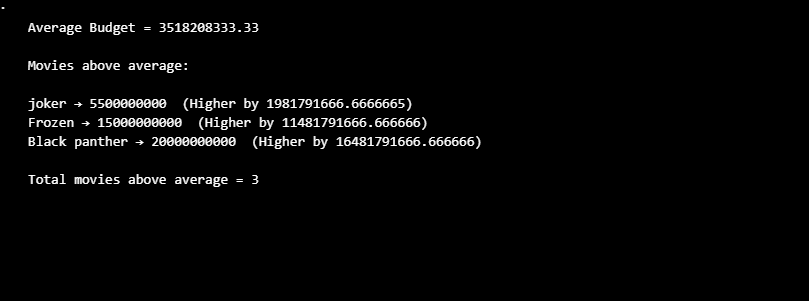
**Why This Approach:**

This method is easy to implement and understand. Using a class makes the code organized. By calculating the average first, we can easily compare each movie budget with it. The approach is flexible and also allows the user to add more movies if required.

**Example Movies:**

Titanic – 200000000  
The Dark Knight – 185000000  
Joker – 55000000  
Frozen II – 150000000  
Black Panther – 200000000

**Output:**

****