Assignment Questions:

a. Python basics

1. **Math Operations:**
    - Create a program that calculates the area of a circle given its radius.
    - Find Armstrong number in range (0,1000)
    - Check where the given number is prime or not?
2. **String Manipulation:**
    - Implement a program that counts the number of vowels in a given string.
    - Create a program that checks if a given word, is a palindrome.
3. **List and Looping:**
    - Write a one-liner using list comprehension to generate a list of squares of numbers from 1 to 10.
    - Implement a function that takes a list of temperatures in Celsius and uses the **map** function to convert them to Fahrenheit.
4. **Conditional Statements:**
    - Implement a program that checks if a given number is even or odd without modular operator.
    - Write a program that determines whether a year is a leap year or not.
5. **Logical Patterns:**
    - Create a program to print a right-angled triangle pattern using asterisks.
    - Write a program to print the Fibonacci sequence up to a specified number of terms.
    - Implement a program to generate the following pattern:

        4

        4 3

        4 3 2

        4 3 2 1
6. **File Handling:**
    - Write a program to read data from a file and display it on the console.
    - Create a program that writes a list of numbers to a pickle file.
7. **Random Module:**
    - Implement a program that generates a random number between a specified range.
    - Write a program that simulates a dice roll.
8. **Function Definitions:**
    - Create a function to calculate the factorial of a list of numbers.

- Write a function that returns the greatest common divisor (GCD) of two numbers.
- Solve prime number using recursion.

9. **Dictionary Manipulation:**
    - Write a program that counts the frequency of each character in a given string and stores it in a dictionary.
    - Create a program that combines two dictionaries into a new one.
    - Update, remove, add operations sort, threshold filtering.
10. **Error Handling:**
    - Implement a program that handles exceptions for division by zero.
    - Write a program that prompts the user to enter a number and handles the Value Error if the input is not a valid number.

b. Advance Python

❖ **Generators:**

1. **Fibonacci Generator:**
    - Implement a generator that generates the Fibonacci sequence up to a specified limit.

❖ **Decorators:**

2. **Timing Decorator:**
    - Implement a decorator that calculates and prints the execution time of a function.
3. **Memorization Decorator:**
    - Create a decorator for memorization, which stores the results of expensive function calls and returns the cached result when the same inputs occur again.
4. **Authentication Decorator:**
    - Implement a decorator that checks if a user is authenticated before allowing access to a specific function.

❖ **Context Managers:**

5. **Timer Context Manager:**
    - Develop a context manager that measures and prints the time taken for the code block inside it to execute.
6. **File Handling Context Manager:**
    - Create a context manager for file handling that ensures proper opening and closing of a file.

❖ **Object-Oriented Programming:**

7. **Bank Account Class:**
    - Design a **Bank Account** class that includes methods for deposit, withdrawal, and checking the balance.
8. **Inheritance:**
    - Create a base class **Shape** and derived classes **Circle** and **Rectangle**. Include methods to calculate the area and perimeter for each shape.
9. **Composition:**
    - Design a class representing a **Library** that contains books. Each book can have multiple authors. Use composition to represent the relationship between the library and books.
10. **Polymorphism:**
    - Implement a class hierarchy for different types of animals. Use polymorphism to create a function that prints the sound each animal makes.