1. Write a Python program to create a class representing a Circle. Include methods to calculate its area and perimeter.

2. Write a Python program to create a person class. Include attributes like name, country and date of birth. Implement a method to determine the person's age.

3. Write a Python program to create a calculator class. Include methods for basic arithmetic operations.

4. Write a Python program to create a class that represents a shape. Include methods to calculate its area and perimeter. Implement subclasses for different shapes like circle, triangle, and square.

5. Write a Python program to create a class representing a binary search tree. Include methods for inserting and searching for elements in the binary tree.

6. Write a Python program to create a class representing a stack data structure. Include methods for pushing and popping elements.

7. Write a Python program to create a class representing a linked list data structure. Include methods for displaying linked list data, inserting and deleting nodes.

8. Write a Python program to create a class representing a shopping cart. Include methods for adding and removing items, and calculating the total quantity.

9. Write a Python program to create a class representing a stack data structure. Include methods for pushing, popping and displaying elements.

10. Write a Python program to create a class representing a queue data structure. Include methods for enqueueing and dequeuing elements

11. Write a python program to create a class representing a bank. Include methods for managing customer accounts and transactions.

12. Create a Class with instance attributes. Write a Python program to create a Vehicle class with max_speed and mileage instance attributes.

13. Create a child class Bus that will inherit all of the variables and methods of the Vehicle class.

14. Class Inheritance

Given:

Create a **Bus** class that inherits from the **Vehicle** class. Give the capacity argument of Bus.seating_capacity() a **default** value of 50.

Use the following code for your parent Vehicle class.

```
class Vehicle:

    def __init__(self, name, max_speed, mileage):

        self.name = name

        self.max_speed = max_speed

        self.mileage = mileage



    def seating_capacity(self, capacity):

        return f"The seating capacity of a {self.name} is {capacity} passengers"
```